



**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
FACULDADE DE ENGENHARIA DA COMPUTAÇÃO E TELECOMUNICAÇÕES**

**Avaliação de métodos de Aprendizado de Máquina
não supervisionados na detecção de ataques de negação de serviço na
Internet das Coisas (IoT).**

DIEGO MEDEIROS DE ABREU

**BELÉM - PA
2018**



**UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
FACULDADE DE ENGENHARIA DA COMPUTAÇÃO E TELECOMUNICAÇÕES**

DIEGO MEDEIROS DE ABREU

**Avaliação de métodos de Aprendizado de Máquina
não supervisionados na detecção de ataques de negação de serviço na
Internet das Coisas (IoT).**

Trabalho de Conclusão de Curso
apresentado para obtenção do grau de
Engenheiro da Computação do Instituto de
Tecnologia da Faculdade de Engenharia da
Computação e Telecomunicações.

**Belém - PA
2018**



SERVIÇO PÚBLICO FEDERAL
UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
FACULDADE DE ENGENHARIA DA COMPUTAÇÃO E TELECOMUNICAÇÕES

ATA da sessão de defesa de trabalho de conclusão de curso da Faculdade de Engenharia da Computação e Telecomunicações do Instituto de Tecnologia da Universidade Federal do Pará.

Ao décimo oitavo dia do mês de dezembro do ano de dois mil e dezoito, às nove horas e trinta minutos, reuniram-se no Laboratório de Tecnologias de Informação e Comunicação, na Universidade Federal do Pará, em sessão pública, os Professores Antônio Jorge Gomes Abelém, Marcos César da Rocha Seruffo, Clodomiro de Souza de Sales Junior e Igor Furtado Carvalho, para formarem a banca examinadora da defesa de trabalho de conclusão de curso do graduando **Diego Medeiros de Abreu**, cujo trabalho “**Avaliação de métodos de Aprendizado de Máquina não supervisionados na detecção de ataques de negação de serviço na Internet das Coisas (IoT)**”, sendo orientado pelo Professor Antônio Jorge Gomes Abelém e coorientado pelo Igor Furtado Carvalho que iniciou a reunião, descrevendo sumariamente o rito desta; apresentou os avaliadores e o autor do trabalho, a quem foi dado um intervalo de tempo de até trinta minutos para desenvolver a apresentação de seu trabalho de conclusão de curso. Encerrada a defesa, o discente foi arguido oralmente pelos examinadores. Foi concedida a oportunidade aos presentes para se pronunciarem. Em seguida, a banca reuniu-se em caráter sigiloso para decidir o resultado do exame, ato que deliberou a aprovação do trabalho. Foi recomendada a aprovação de **Diego Medeiros de Abreu** no trabalho de conclusão de curso, por ter atendido aos requisitos solicitados, que foram lavrados em folhas próprias para guardar na Secretaria do Curso. Observadas as alterações sugeridas, o discente foi orientado a apresentar a versão final de seu trabalho de conclusão de curso no prazo de até trinta dias. Nada mais havendo a tratar, a sessão foi encerrada, lavrando-se dela a presente ata, que uma vez aprovada, foi assinada pelos membros da banca examinadora e pelo discente. Belém do Pará, dezoito de dezembro de dois mil e dezoito.

Prof. Dr. Antônio Jorge Gomes Abelém
Presidente/Orientador

Prof. Dr. Marcos César da Rocha Seruffo
Membro

Prof. Clodomiro de Souza de Sales Junior
Membro

Igor Furtado Carvalho
Coorientador

Diego Medeiros de Abreu
Autor do Trabalho

DEDICATÓRIA

Dedico esse trabalho a Deus que sempre esteve comigo, aos meus pais , aos meus familiares e amigos.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela sua fidelidade em ter sustentando minha caminhada até aqui e ter se mostrado presente em todos os percalços.

Agradeço aos meus pais, Jonas Gomes de Abreu e Gecilêda Medeiros de Abreu, e minha irmã, Bianca Medeiros de Abreu, pelo amor, cuidado, educação e apoio investidos a mim.

Agradeço também à minha namorada Karla Jacqueline Pinto de Lima, pelo amor e auxílio durante a escrita deste trabalho.

Agradeço por fim ao meu orientador, Antônio Jorge Gomes Abelém, e ao meu coorientador, Igor Furtado Carvalho, e toda a equipe do GERCOM-UFPa, por me proporcionarem a oportunidade e me guiarem durante a jornada do conhecimento acadêmico.

O temor do Senhor é o princípio do saber.

(Provérbios 9:10a)

EPÍGRAFE

RESUMO

Nos últimos anos ocorreu um grande aumento na quantidade e na variedade de dispositivos da Internet das Coisas, como televisores, câmeras e relógios inteligentes. Porém, o crescimento desses dispositivos tem sido visto como um problema de segurança na rede e muitos dispositivos IoT estão sendo utilizados para realizar ataques, como ataques de negação de serviço em grande escala.

Nesse contexto, este trabalho de conclusão de curso propõe-se a avaliar a viabilidade da abordagem de aprendizado de máquina não supervisionado na detecção de ataques de agentes maliciosos em redes domésticas com dispositivos IoT. Os dados utilizados para análise dos algoritmos serão reais e gerados em cenários com a presença de dispositivos IoT.

Assim, esse estudo irá indicar os métodos de aprendizagem de máquina não supervisionado e as informações dos pacotes de redes mais adequados para detectar ação desses agentes maliciosos, tornando possível diagnosticar mais rapidamente esses ataques na rede.

Palavras-chave: Internet das Coisas; Aprendizado de Máquina; Ataque de Negação de Serviço.

ABSTRACT

In recent years there has been a large increase in the quantity and variety of Internet of Things devices such as smart TV's, cameras and smart watches. However, the influx of these devices has been seen as a security issue as many IoT devices are being used to conduct network attacks, such as large-scale denial of service attacks. In this context, this final paper aims to evaluate the feasibility of the unsupervised machine learning approach in detecting attacks of malicious agents in home networks with IoT devices. The data used to analyze the algorithms will be real and generated in scenarios with the presence of IoT devices. Therefore, this study will indicate the most suitable unsupervised machine learning methods and the most suitable network packet features to detect the action of these malicious agents.

Keywords: Internet of Things; Machine Learning; Denial of Service Attack.

LISTA DE FIGURAS

Figura 1 Estimativa do número de aparelhos conectados à Internet	22
Figura 2 Crescimento do número de dispositivos conectados com relação a população mundial de pessoas.	22
Figura 3 Subdivisões do Aprendizado de Máquina.	25
Figura 4 Aprendizado Supervisionado e Não Supervisionado.....	25
Figura 5 Comparação entre as distâncias de manhattan e euclidiana.....	28
Figura 6 Agrupamento CANOPY.....	31
Figura 7 Funcionamento do handshake de três vias do protocolo TCP....	32
Figura 8 Funcionamento do ataque SYN Flood.	33
Figura 9 Funcionamento do ataque UDP flood.	34
Figura 10 Funcionamento do ataque ICMP flood.....	35
Figura 11 Funcionamento da Botnet Mirai: seus componentes e etapas. ...	36
Figura 12 Fluxograma da classificação via clusterização.	41
Figura 13 Arquitetura dos experimentos.	43

LISTA DE TABELAS

Tabela 1 Matriz de confusão.....	38
Tabela 2 Descrição dos Cenários.....	44
Tabela 3 Resultados dos algoritmos com relação a performance das métricas acertos utilizando Tamanho do Pacote, Protocolo, Tempo entre chegada dos Pacotes para a Base de Dados 1.....	47
Tabela 4 Resultados dos algoritmos com relação à matriz de confusão utilizando Tamanho do Pacote, Protocolo, Tempo entre chegada dos pacotes para a Base de Dados 1.....	47
Tabela 5 Resultados dos algoritmos com relação a performance das métricas acertos, utilizando apenas o Tamanho dos Pacotes como atributo, para a Base de Dados 1.....	49
Tabela 6 Resultados dos algoritmos com relação à matriz de confusão, utilizando apenas o Tamanho dos Pacotes como atributo, para a Base de Dados 1.....	49
Tabela 7 Resultados dos algoritmos com relação a performance das métricas acertos, utilizando apenas o Protocolo como atributo, para a Base de Dados 1.....	50
Tabela 8 Resultados dos algoritmos com relação à matriz de confusão, utilizando apenas o Protocolo como atributo, para a Base de Dados 1.....	51
Tabela 9 Resultados dos algoritmos com relação a performance das métricas acertos, utilizando apenas o Tempo entre chegada dos Pacotes como atributo para a Base de Dados 1.....	52
Tabela 10 Resultados dos algoritmos com relação à matriz de confusão, utilizando apenas o Tempo entre chegada dos Pacotes como atributo para a Base de Dados 1.....	52
Tabela 11 Resumo Comparativo dos Resultados para a Base de Dados 1.....	54
Tabela 12 Resultados dos algoritmos com relação a performance das métricas acertos, utilizando o Protocolo, Tamanho dos pacotes e o Tempo entre chegada dos Pacotes como atributo para a Base de Dados 1.....	56

Tabela 13 Resultados dos algoritmos com relação à matriz de confusão, utilizando o Protocolo, Tamanho dos Pacotes e o Tempo entre chegada dos Pacotes como atributo para a Base de Dados 2.....	56
Tabela 14 Resultados dos algoritmos com relação a performance das métricas acertos, utilizando apenas o Tamanho dos Pacotes como atributo para a Base de Dados 2	58
Tabela 15 Resultados dos algoritmos com relação à matriz de confusão, utilizando apenas o Tamanho dos Pacotes como atributo para a Base de Dados 2.....	58
Tabela 16 Resultados dos algoritmos com relação a performance das métricas acertos, utilizando apenas o Protocolo como atributo para a Base de Dados 2.....	59
Tabela 17 Resultados dos algoritmos com relação à matriz de confusão, utilizando apenas o Protocolo como atributo para a Base de Dados 2.....	59
Tabela 18 Resultados dos algoritmos com relação a performance das métricas acertos, utilizando apenas o Tempo entre chegada dos Pacotes como atributo para a Base de Dados 2.....	60
Tabela 19 Resultados dos algoritmos com relação à matriz de confusão, utilizando apenas o Tempo entre chegada dos Pacotes como atributo para a Base de Dados 2.	61
Tabela 20 Resumo Comparativo dos Resultados para a Base de Dados 2.....	62
Tabela 21 Comparativo dos Resultados para a Base de Dados 1 e 2.....	64

LISTA DE EQUAÇÕES

Equação 1 Fórmula do Algoritmo K-Means.....	28
Equação 2 Fórmula da Distância Euclidiana.....	28
Equação 3 Fórmula da Distância de Manhattan.	28
Equação 4 Fórmula da Acurácia	39
Equação 5 Fórmula da Precisão	40
Equação 6 Fórmula da Medida F1	40

LISTA DE ALGORITMOS

Algoritmo 1 Algoritmo K-Means.....	27
Algoritmo 2 Algoritmo do Expectation-Maximization (EM)	29
Algoritmo 3 Algoritmo do DBSCAN.....	30

LISTA DE ABREVIATURAS E SIGLAS

DoS	Ataque de Negação de Serviço
DDoS	Ataque Negação de Serviço Distribuído
FP	Falso Positivo
FN	Falso Negativo
ICMP	Protocolo de Mensagens de Controle da Internet
IoT	Internet das Coisas
RFID	Identificador de Rádio Frequência
TCP	Protocolo de Controle de Transmissão
UDP	Protocolo de Datagrama de Usurário
VP	Verdadeiro Positivo
VN	Verdadeiro Negativo.

SUMÁRIO

1 INTRODUÇÃO	19
1.1 OBJETIVOS.....	20
1.1.1 Objetivo Geral	20
1.1.2 Objetivos Específicos	20
1.2 ORGANIZAÇÃO DO TRABALHO	20
2 EMBASAMENTO TEÓRICO.....	21
2.1 INTERNET DAS COISAS.....	21
2.2 APRENDIZADO DE MÁQUINA	24
2.2.1 Aprendizado de Máquina Supervisionado e Não Supervisionado	25
2.3 ALGORITMOS NÃO SUPERVISIONADOS	27
2.3.1 Método K-Means	27
2.3.2 Expectation-Maximization.....	29
2.3.3 DBSCAN	30
2.3.4 Farthest First.....	30
2.3.5 CANOPY	31
2.4 ATAQUES DE NEGAÇÃO DE SERVIÇO	32
2.4.1 SYN Flood.....	32
2.4.2 UDP Flood	34
2.4.3 ICMP Flood	35
2.4.4 Ataque Mirai	36
2.5 MÉTRICAS DE DESEMPENHO.....	38
2.6 CLASSIFICAÇÃO POR CLUSTER	41
2.7 TRABALHOS RELACIONADOS	42
3 METODOLOGIA- ESTUDO DE CASO	43
3.1 ARQUITETURA DO EXPERIMENTO	43
3.2 CENÁRIOS DO EXPERIMENTO	44

3.3 AVALIAÇÃO DE DESEMPENHO.....	46
4 RESULTADOS.....	47
4.1. RESULTADOS UTILIZANDO A BASE DE DADOS 1	47
4.1.1 Resultados utilizando Tamanho do Pacote, Protocolo e Tempo entre chegada dos Pacotes, para a base de dados 1.	47
4.1.2 Resultados utilizando apenas o Tamanho dos Pacotes como atributo, para a Base de Dados 1.....	49
4.1.3 Resultados utilizando apenas o Protocolo como atributo, para a Base de Dados 1.....	50
4.1.4 Resultados utilizando apenas o Tempo entre chegada dos Pacotes como atributo, para a Base de Dados 1.	52
4.1.5 Conclusão dos resultados para a Base de Dados 1	54
4.2. RESULTADOS UTILIZANDO A BASE DE DADOS 2	56
4.2.1 Resultados utilizando Protocolo, Tamanho dos Pacotes e Tempo entre chegada dos Pacotes como atributo, para a Base de dados 2.	56
4.2.2 Resultados utilizando apenas o Tamanho dos Pacotes como atributo, para a Base de Dados 2.....	58
4.2.3 Resultados utilizando apenas o Protocolo como atributo, para a Base de Dados 2.	59
4.2.4 Resultados utilizando apenas o Tempo entre chegada dos Pacotes como atributo, usando a Base de Dados 2.	60
4.2.5 Conclusão dos resultados para a Base de Dados 2	62
4.3 CONCLUSÃO DOS RESULTADOS.....	64
5. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....	66
REFERÊNCIAS.....	67

1. INTRODUÇÃO

Nos últimos anos ocorreu um grande aumento na quantidade e na variedade de dispositivos da Internet das Coisas (IoT), como *smart* TV's, câmeras e relógios inteligentes. O crescimento desses dispositivos tem sido visto como um problema de segurança na rede, já que, como aponta um relatório recente da Kaspersky Lab, muitos dispositivos IoT estão sendo utilizados para realizar ataques como DoS em grande escala (COSSETTI, 2018).

O estudo da detecção de ataques, como ataques de negação de serviços em redes domésticas, já foi amplamente discutido em diferentes estudos acadêmicos como CHEN et al. (2013) e JUN et al. (2011). Entretanto, as propostas apresentadas geralmente são baseadas em modelos com pouca presença de dispositivos IoT, os quais, como aponta JYOTHI et al. (2016), possuem comportamento muito diferente dos dispositivos mais tradicionais, como computadores e notebooks. Além disso, muitos artigos como PERVEZ (2014), utilizam dados que resultam de simulação ou base de dados antigas, e não dados obtidos em experimentos com redes reais, o que pode distanciar os dados coletados da realidade. Por fim, pesquisas como DOSHI et al. (2018) geralmente se focam apenas no desempenho de métodos supervisionados.

Este trabalho de conclusão de curso faz parte de um projeto de pesquisa que busca encontrar formas de melhorar a segurança em redes domésticas com a presença de dispositivos IoT. Um dos grandes desafios atuais é realizar a detecção de ataques de negação de serviços, principalmente dos chamados “zero-day attacks”, os quais são ataques novos e que mudam constantemente de comportamento. Nesse contexto, tem-se estudado a utilização de técnicas de Aprendizado de Máquina, tanto de métodos supervisionados quanto não supervisionados para realizar a detecção desses ataques.

Desse modo, neste trabalho de conclusão de curso, será avaliada a viabilidade do uso de métodos e algoritmos de Aprendizado de Máquina não supervisionado, para a detecção de ataques de negação de serviço na rede. Assim, esse estudo irá indicar os métodos não supervisionados mais adequados para detectar ação desses agentes maliciosos, tornando possível diagnosticar mais rapidamente a ocorrência desses ataques de negação de serviço.

1.1. OBJETIVOS

1.1.1. Objetivo Geral

Avaliar a viabilidade da abordagem de aprendizado de máquina não supervisionada na detecção de ataques de agentes maliciosos em redes IoT, avaliando e comparando o desempenho dos diversos métodos e algoritmos não supervisionados disponíveis.

1.1.2. Objetivos Específicos

1. Avaliar o comportamento dos algoritmos de aprendizado de máquina em diferentes cenários de ataques e comparar os resultados obtidos para cada cenário.
2. Avaliar o comportamento das características do pacotes de rede (Tamanho do Pacote, Tempo entre chegada do Pacote e Protocolo) e comparar o seu desempenho como atributos para o processo de aprendizado de máquina não supervisionado.

1.2. ORGANIZAÇÃO DO TRABALHO

Esse trabalho de conclusão de curso está organizado da seguinte forma. No Capítulo 2 é apresentado o embasamento teórico para o trabalho. Nesse capítulo, são apresentados os tópicos de Internet das Coisas, Aprendizado de Máquina, os Ataques de Negação de Serviço, Métricas de desempenho e Trabalhos Relacionados.

O Capítulo 3 descreve a metodologia aplicada neste trabalho. Serão apresentados a Arquitetura do Experimento, o Cenário de Ataques e a Avaliação de Desempenho para os experimentos realizados.

No Capítulo 4 são apresentados os resultados obtidos para a proposta deste trabalho de conclusão de curso. Nesse capítulo, serão apresentados os resultados para cada um das base de dados, bem como são discutido a performance dos algoritmos e da metodologia não supervisionada com base nas métricas de desempenho estudadas. Por fim, o Capítulo 5 contém as considerações finais e as sugestões de trabalhos futuros.

2. EMBASAMENTO TEÓRICO

2.1. INTERNET DAS COISAS

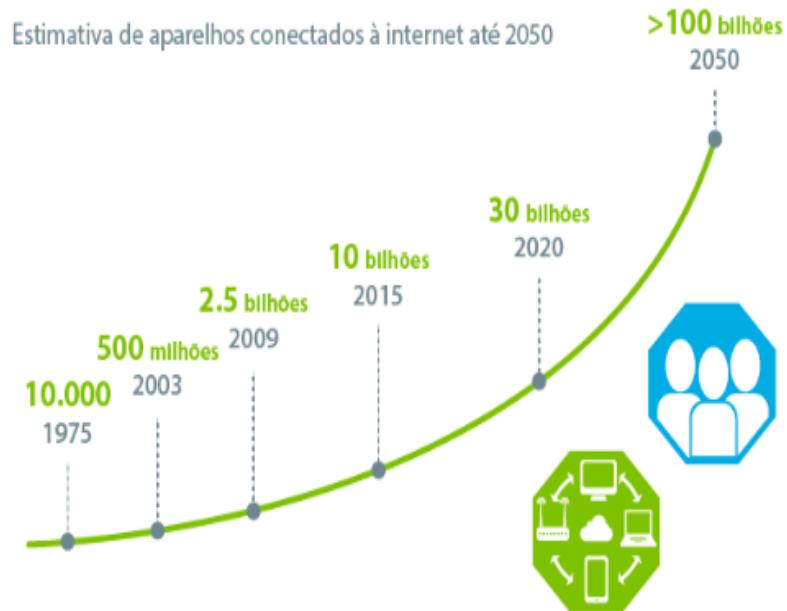
Em 1999 Kevin Ashton, propôs o termo Internet das Coisas para descrever como os objetos no mundo físico se conectam à Internet, e como funcionava o sistema de etiquetas eletrônicas, baseadas em RFID, dos produtos de uma linha de produção. Com base nessa ideia, percebeu-se a oportunidade de comunicação direta entre dispositivos, de modo que pudessem se comunicar entre si. (ASHTON,2009)

Segundo BENGHOZI et al.(2009), a Internet das Coisas pode ser definida como: “uma rede de redes que permite a identificação de entidades digitais e objetos físicos - sejam inanimados (incluindo plantas) ou animados (animais e seres humanos) - diretamente e sem ambiguidade, através de sistemas padronizados de identificação eletrônica dispositivos, e assim possibilitar a recuperação, armazenamento, transferência e processamento de dados relacionados a eles, sem descontinuidade entre os mundos físico e virtual”

De outro modo, os objetos físicos (as coisas) são os elementos que possuem capacidade de comunicação e de processamento. Esses dispositivos não são apenas computadores convencionais, mas também podem ser televisores, notebooks, automóveis, câmeras, sensores ou qualquer equipamento que possua uma forma de conexão à rede. Como afirma BENGHOZI et al.(2009), os objetos da Internet das coisas são “Coisas com identidades e personalidades virtuais operando em espaços inteligentes, usando interfaces inteligentes para se conectar e se comunicar dentro de contextos sociais, ambientais e de usuários”.

Na medida em que o acesso ao serviço de Internet de banda larga cresce e os processadores se tornam mais acessíveis, e mais dispositivos com recursos de Wi-Fi são criados. A Figura 1 apresenta uma previsão do aumento de dispositivos conectados à Internet.

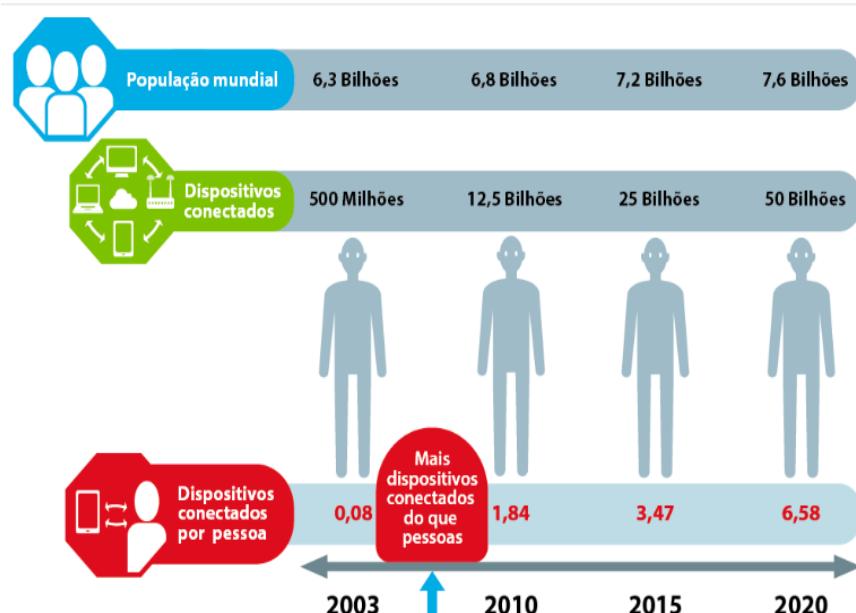
Figura 1 - Estimativa do número de aparelhos conectados à Internet.



Fonte: Community IoT One (2018).

Como pode ser visto na Figura 1, o número de dispositivos que estarão conectados à Internet tem crescido muito nos últimos anos e esse crescimento tende a aumentar com a presença dos dispositivos IoT. A Figura 2 coloca em perspectiva a relação entre dispositivos conectados a Internet com a população mundial.

Figura 2- Crescimento do número de dispositivos conectados com relação a população mundial de pessoas.



Fonte: EVANS (2018).

Como a Figura 2 representa, atualmente já existem mais dispositivos conectados do que pessoas, e esse número só tende a aumentar de modo que existe essa tendência de que a Internet atual cada vez mais se torne uma Internet das Coisas.

No entanto, a crescente popularidade da Internet das Coisas fez dos dispositivos IoT uma poderosa plataforma de amplificação para ataques cibernéticos. Muitos desses dispositivos IoT são fundamentalmente inseguros. Como exemplo disso, temos o impacto dos ataques DDoS pelo *malware* Mirai, suas variantes e outras ataques similares, os quais destacam os riscos que os dispositivos IoT representam para a Internet.

Segundo DOSHI et al. (2018) , existem cinco razões principais pelas quais os dispositivos IoT são particularmente vantajosos para difusão de ataques DDoS:

1. Operação constante: Ao contrário dos computadores, laptop e desktop, que têm ciclos *on-off* frequentes, ou seja, que costumam ser ligados e desligados frequentemente, muitos dispositivos IoT, como webcams e roteadores sem fio, costumam operar sem serem desligados por muito tempo. Assim, os dispositivos IoT tem um maior tempo de disponibilidade para serem utilizados nos ataques DDoS.
2. Sistema de segurança com pouca proteção: Segundo DOSHI et al. (2018), Muitos fornecedores de dispositivos negligenciam a segurança dos dispositivos IoT, em favor da facilidade de uso pelo usuário. Desse modo, poucos aparelhos possuem algum sistema de segurança previsto, como um antivírus ou algum protocolo de segurança que possa impedir a ação de agentes maliciosos através da rede.
3. Manutenção do dispositivo é rara ou ocasional : A maioria dos dispositivos IoT acabam dentro do costume de configurar e esquecer. Depois de instalá-los e configurá-los inicialmente, os usuários e os próprios administradores de rede geralmente os esquecem, a menos que o dispositivo pare de funcionar corretamente, o que geralmente só ocorre quando ação dos agentes maliciosos já é intensa.
4. Possível tráfego de rede gerado para ataques é considerável: Os dispositivos de IoT são capazes de produzir tráfego de rede em ataque DDoS comparável ao dos sistemas de desktop modernos. Assim, os dispositivos IoT podem ser

tão úteis para um ataque DDoS quanto um computador desktop comum em redes domésticas atuais.

5. Algumas interfaces de usuário são minimamente interativas: Como os dispositivos de IoT geralmente tendem a exigir uma intervenção mínima do usuário, é mais provável que as infecções de *malwares* como o Mirai passem despercebidas pelo usuário.

Se a comunidade de segurança computacional não responder mais rapidamente, os ataques cibernéticos que utilizam dispositivos IoT se tornarão a norma e poderão comprometer a própria infraestrutura da Internet. Nesse contexto, formas de segurança de rede utilizem uma metodologia de aprendizado de máquina tem surgido como uma forma de realizar a detecção desses ataques nas redes IoT e reduzir o impacto que os agentes maliciosos possam causar.

2.2. APRENDIZADO DE MÁQUINA

O Aprendizado de Máquina, como descrito em SAMUEL (1959) é um "campo de estudo que dá aos computadores a capacidade de aprender sem serem programados explicitamente". O Aprendizado de Máquina tem como objetivo o desenvolvimento de técnicas computacionais capazes de adquirir conhecimento de forma automática. (SHANTHAMALLU,2017).

Em MITCHELL (1997) o aprendizado de máquina é definido como quando: "*Um programa de computador é dito que aprende com a experiência em relação a um conjunto de classes de tarefa e a medida de desempenho, se seu desempenho em tarefas, tal como medido, melhora com a experiência*". Isto é, um algoritmo de aprendizado de máquina, de forma geral, deve melhorar de desempenho conforme recebe uma experiência vinda de um conjunto de dados.

Assim, temos que o aprendizado é o processo de converter a experiência em conhecimento. A entrada para um algoritmo de aprendizado é o treinamento de dados, representando a experiência, e a saída é alguma especialização, que geralmente toma a forma de outro programa de computador que pode executar alguma tarefa.

2.2.1. Aprendizado de Máquina Supervisionado e Não Supervisionado.

Tipicamente os algoritmos de Aprendizado de Máquina (AM) são classificados em Supervisionados e Não Supervisionado (Shalev-Shwartz e Ben-David, 2014). A Figura 3 ilustra essa divisão.

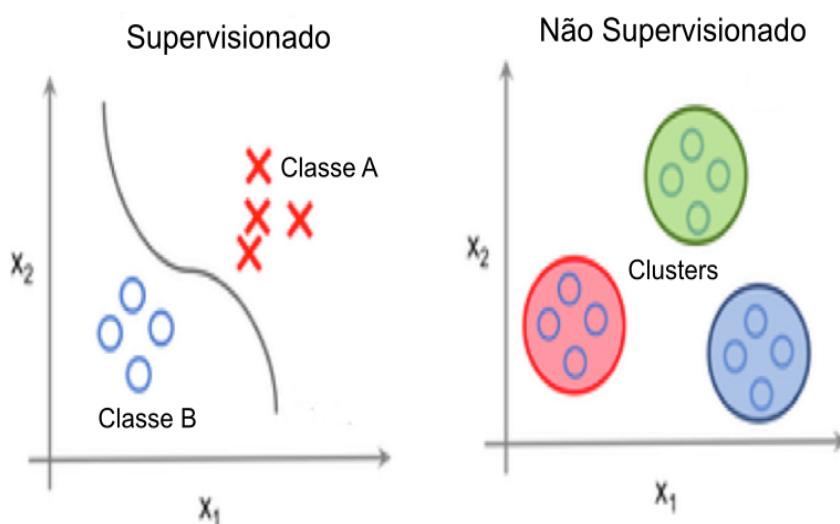
Figura 3- Subdivisões do Aprendizado de Máquina.



Fonte: ARAÚJO (2015).

O aprendizado de máquina supervisionado é baseado na predição, ou seja, determina com base em padrões uma previsão ou tendência em informações ausentes ou desconhecidas. Para isso, são apresentadas ao computador exemplos de entradas e saídas desejadas. O objetivo desse tipo de metodologia é aprender uma regra geral que mapeia as entradas para as saídas. A Figura 4 ilustra o aprendizado supervisionado e não supervisionado.

Figura 4- Aprendizado Supervisionado e Não Supervisionado



Fonte: Adaptado de KURAMA (2015).

Na aprendizagem supervisionada, estão disponíveis rótulos verdadeiros ou corretos do conjunto de dados de entrada. O algoritmo é treinado usando o conjunto de dados de entrada rotulado (dados de treinamento), o que significa que as amostras contendo as informações reais estão disponíveis para treinamento.

No processo de treinamento, o algoritmo faz previsões apropriadas sobre os dados de entrada e melhora suas estimativas até que o algoritmo atinja um nível desejado de precisão. Como exemplo da classificação supervisionada, temos os métodos de classificação, como árvores de decisão e redes neurais, e métodos de regressão, como a regressão linear e a regressão logística.

De outro modo, os Algoritmos de aprendizado não supervisionados são projetados para descobrir estruturas ocultas em conjuntos de dados não rotulados, nos quais a saída desejada é desconhecida. Como exemplo de Aprendizado de máquina não-supervisionado, temos os métodos de cluster, que serão utilizados para realização desta pesquisa.

A clusterização (*clustering*) é um conjunto de técnicas para encontrar padrões em dados não rotulados de alta dimensionalidade. É uma abordagem de aprendizado de máquina não supervisionada em que os dados são agrupados com base em uma medida de similaridade.

As técnicas de clusterização buscam encontrar uma estrutura ou padrão em uma coleção de conjunto de dados. Para um dado conjunto de dados, o algoritmo agrupa os dados fornecidos em certo número de clusters, de modo que os pontos dos dados dentro de cada cluster sejam semelhantes entre si e os pontos de dados de clusters diferentes sejam diferentes entre si.

A principal vantagem do *clustering* para detecção de intrusão é que essa metodologia pode aprender com os dados de entrada , sem exigir que um supervisor forneça descrições explícitas do vários tipos de ataques, o que possibilita a detecção de novos ataques de intrusão. (BUCZARK E GUVEN, 2016)

2.3. ALGORTIMOS NÃO SUPERVISIONADOS

2.3.1. Método K-Means

O agrupamento K-Means é um método de clusterização que tem como objetivo dividir os dados de entrada entre k grupos, no qual cada dado pertence ao grupo mais próximo da média da posição dos dados.

A ideia do algoritmo K-Means é fornecer uma classificação de informações de acordo com os próprios dados. Esta classificação é baseada em análise e comparações entre os valores numéricos dos dados. Dessa maneira, o algoritmo automaticamente vai fornecer uma classificação sem a necessidade de nenhuma pré-classificação existente. O Algoritmo 1 apresenta o funcionamento do método K-Means.

Algoritmo 1- Algoritmo do K-Means

Algoritmo 1: Algoritmo k -means básico

início

 Seleciona k pontos como centroides iniciais;

 repita

 Forme k grupos atribuindo cada ponto ao seu centroide mais próximo;

 Recalcule o centroide de cada grupo;

 até que os centroides não mudem;

fim

Fonte: TAN; STEINBACH; KUMAR (2009).

Como o Algoritmo 1 apresenta, o método K-Means inicia com um número k de *clusters* a ser criado, cada um com um valor de centro, os centroides. Após isso, o algoritmo vai analisar todos os dados indicando o cluster mais apropriado para cada um dos dados, com base num fator de proximidade calculado por uma função distância.

Então, o valor do centroide de cada um dos k *clusters* é recalculado e o processo é repetido até que os valores dos centroides não mudem. A Equação 1 apresenta a fórmula para o algoritmo K-Means.

Equação 1- Fórmula do Algoritmo K-Means

$$\text{Função objetivo } J = \sum_{j=1}^k \sum_{i=1}^n \left\| x_i^{(j)} - c_j \right\|^2$$

Número de Clusters Número de Casos
 k n
 Caso i
 Centroide j
 Função Distância

Fonte: Adaptado de Sayad (2018).

A função distância calcula o quanto longe uma ocorrência está da outra. Para o algoritmo K-Means, pode-se tanto utilizar a distância Euclidiana, como na Equação 2, quanto a também pode se utilizar a distância de *Manhattan*, apresentada na Equação 3. A Figura 5 apresenta uma comparação entre essas funções distâncias.

Equação 2- Fórmula da Distância Euclidiana.

$$De = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

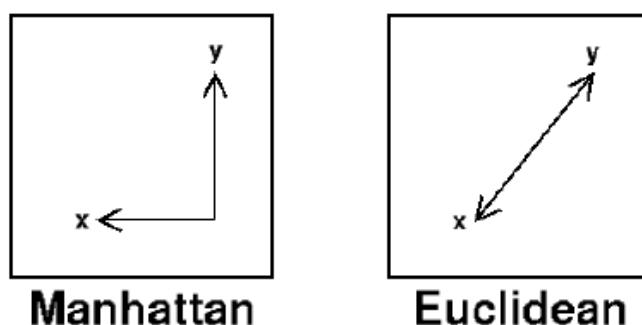
Fonte: Elaborado pelo autor.

Equação 3- Fórmula da Distância de Manhattan.

$$Dm = |X_2 - X_1| + |Y_2 - Y_1|$$

Fonte: Elaborado pelo autor.

Figura 5- Comparação entre as distâncias de Manhattan e Euclidiana.



Fonte: YOUNG (2018) .

Como as equações 2 e 3, e a Figura 5, apresentam, enquanto que a distância Euclidiana é a distância em linha reta entre os dois pontos, a distância de *Manhattan* é a distância que utiliza dois vetores que formam noventa graus entre os dois pontos.

2.3.2. Expectation-maximization

A técnica Expectation-maximization (EM), maximização de expectativa, é semelhante à técnica K-Means. No entanto o algoritmo de clusterização EM calcula as probabilidades de associações de cluster com base em uma ou mais distribuições de probabilidade. O objetivo do algoritmo de agrupamento é, então, maximizar a probabilidade ou a probabilidade geral dos dados. A Algoritmo 2 apresenta o método Expectation-maximization.

Algoritmo 2- Algoritmo do Expectation-Maximization (EM).

Algoritmo 2: Algoritmo EM

início

 Seleciona um conjunto inicial de parâmetros de modelos;
 (Assim como em *k-means*, isto pode ser feito aleatoriamente, em uma diversidade de formas.)

repita

 Etapa da Expectativa Para cada objeto, calcule a probabilidade de que cada objeto pertença a cada distribuição, i.e., calcule $\text{prob}(\text{distribuição } j|x_i, \theta)$;

 Etapa da Maximização Dadas as probabilidades da etapa da expectativa, encontre as novas expectativas dos parâmetros que maximizem a probabilidade esperada;

 até que Os parâmetros não mudem;

 (De forma alternativa, para se a mudança nos parâmetros estiver abaixo de um limite especificado.)

fim

Fonte: TAN; STEINBACH; KUMAR (2009).

2.3.3. DBSCAN

O Método DBSCAN, também conhecido como cluster espacial baseado em densidade de aplicações com ruído, é um algoritmo de *clustering* baseado em densidade. Esse algoritmo encontra um número de clusters a partir da distribuição de densidade estimada dos nós correspondentes. Todos os pontos do cluster são então conectados por densidade mútua. Se um ponto estiver conectado à densidade em qualquer ponto do cluster, ele também faz parte do cluster. O método DBSCAN é apresentado no Algoritmo 3.

Algoritmo 3- Algoritmo do DBSCAN

Algoritmo 3: Algoritmo DBSCAN

```
início
    Rotular todos os pontos como de centro, limite ou ruído;
    Eliminar os pontos de ruído;
    Colocar uma aresta entre todos os pontos de centro que estejam dentro do
        raio  $\epsilon$  uns dos outros;
    Tornar cada grupo de pontos de centro conectados em grupo separado;
    Atribuir cada ponto de limite a um dos grupos dos seus pontos de centro
        associados;
fim
```

Fonte: TAN; STEINBACH; KUMAR (2009).

2.3.4. Farthest First

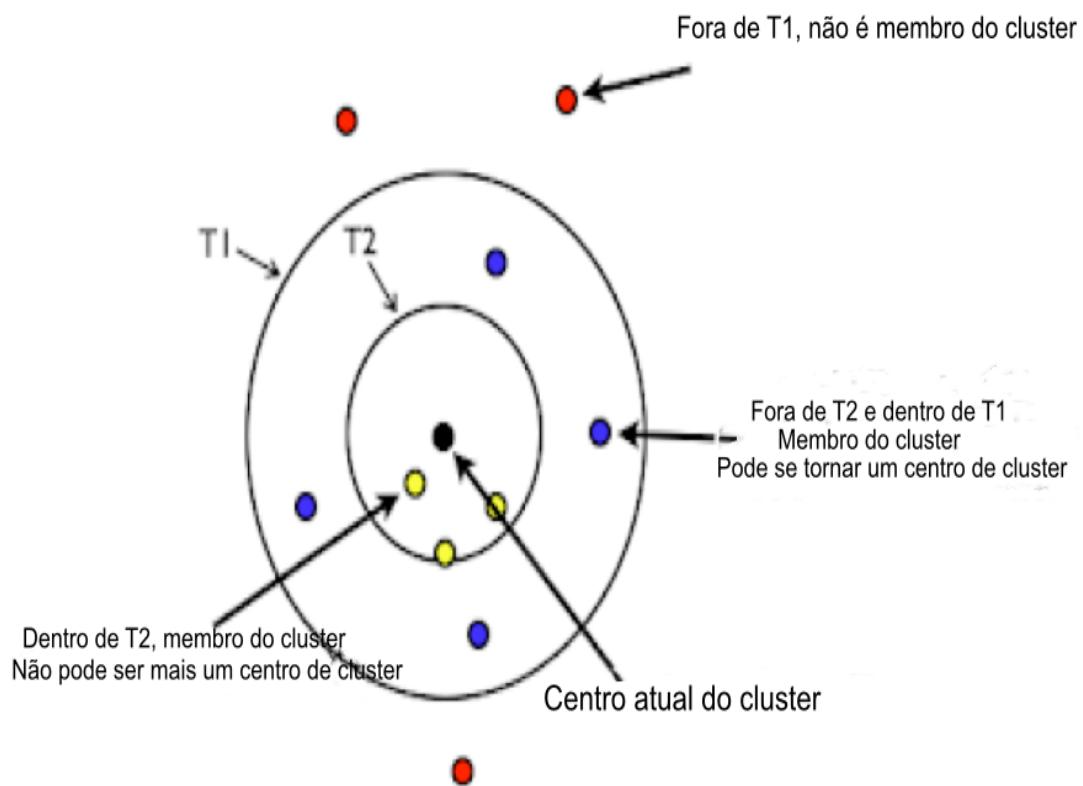
A métrica de *Farthest First* é um sequência de pontos no espaço no qual o primeiro ponto é selecionado arbitrariamente e os pontos posteriores estão o mais distante o possível do último ponto selecionado.

O algoritmo de clusterização *Farthest first*, o mais distante primeiro proposto por HOCHBAUM e SHMOYS (1985) tem funcionamento análogo ao do K-Means. Esse algoritmo escolhe os centroides e designa os dados para os clusters, porém, aloca, inicialmente, os dados nos clusters que estão mais distantes deles.

2.3.5.CANOPY

Segundo MCCALLUM, NIGAM e UNGAR (2000), no método de clusterização Canopy, os subconjuntos criados na primeira fase do método são chamados de canopies. Um Canopy é um agrupamento de elementos formado de acordo com uma medida de distância aproximada, dentro de um limiar mínimo estabelecido, como um índice invertido. Cada elemento pode pertencer a mais de um bloco. A Figura 6 ilustra o agrupamento CANOPY.

Figura 6- Agrupamento CANOPY



Fonte: Adaptado de KUMAR (2013).

Os Canopies seguem a propriedade de que pontos que não estão em um mesmo Canopy são tão distantes que não pode ser possível estarem em um mesmo bloco. Como a distância usada para criar os grupos é aproximada, não se tem como garantir esta propriedade. Mas como os Canopies podem ser sobrepostos através da escolha de um limiar suficientemente grande, na grande maioria dos casos, essa propriedade pode ser garantida.

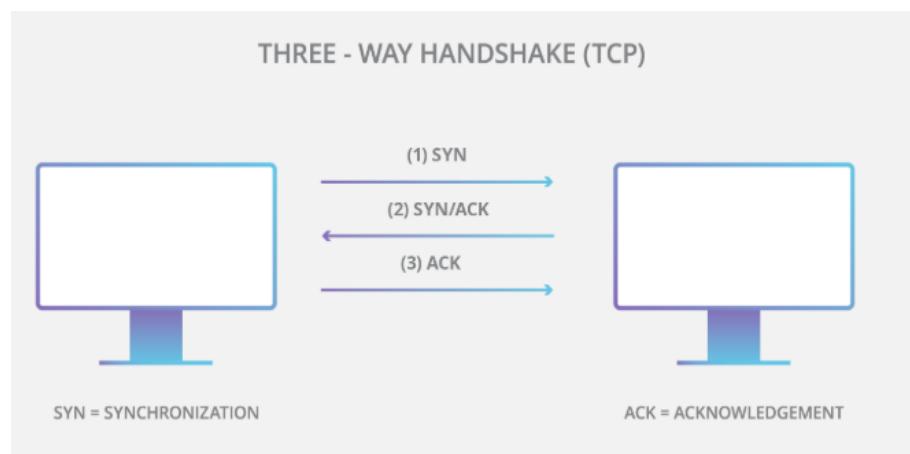
2.4 ATAQUES DE NEGAÇÃO DE SERVIÇO

2.4.1 SYN Flood

Um ataque do tipo *SYN Flood* (inundação SYN) é uma forma de ataque de negação de serviço (DDoS), no qual um invasor envia uma sucessão de solicitações SYN ao sistema de um alvo na tentativa de consumir todos os seus recursos de servidor, isso geralmente é suficiente para tornar o sistema irresponsivo ao tráfego legítimo, de usuários reais.

O ataque *SYN Flood* se aproveita do *handshake* de três vias do protocolo TCP (*three-way handshake*) ao enviar ao sistema alvo múltiplas solicitações do tipo SYN. Essa mensagem indica que uma conexão entre o sistema de origem e o sistema de destino deveria iniciar. A Figura 7 apresenta o funcionamento do *handshake* de três vias do protocolo TCP em uma situação de tráfego normal.

Figura 7- Funcionamento do *handshake* de três vias do protocolo TCP.



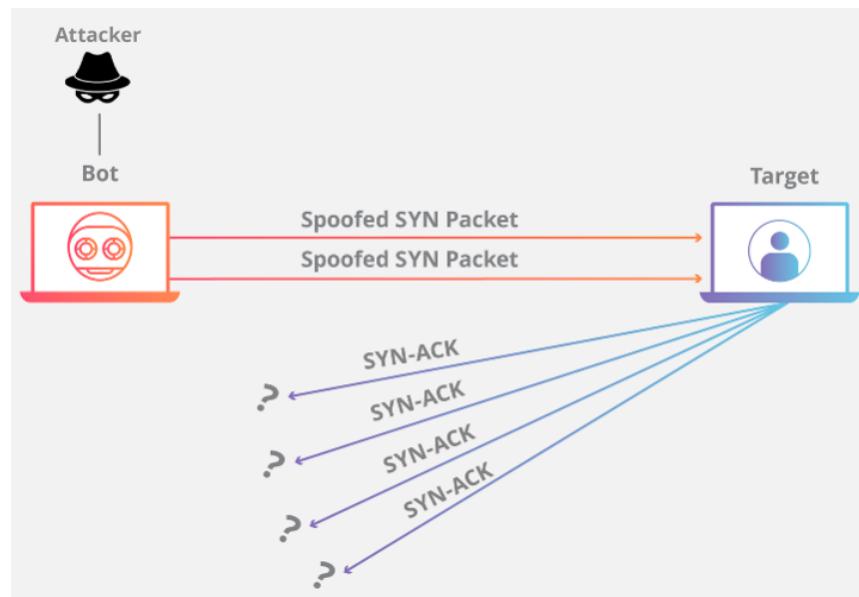
Fonte: Retirado do site cloudflare.com.

1. Primeiro, o cliente envia um pacote SYN para o servidor, indicando o início da conexão.
2. Segundo, o servidor responde àquele pacote inicial com um pacote SYN/ACK, com objetivo de realizar a confirmação da comunicação.
3. Por último, o cliente envia um pacote ACK ao servidor para indicar o recebimento do pacote de confirmação do servidor. Após isso a conexão TCP é finalmente aberta e os dados podem ser enviados/recebidos.

Desse modo, o sistema de destino responde com uma mensagem SYN-ACK (reconhecimento de sincronização) para cada mensagem SYN que recebe e abre temporariamente uma porta de comunicações para cada tentativa de conexão enquanto espera por uma mensagem ACK (confirmação) final do sistema de origem em resposta a cada uma das mensagens SYN-ACK.

Como a Figura 8 apresenta, a fonte de ataque nunca envia as mensagens ACK finais e, portanto, a conexão nunca é concluída. O servidor aguardará o reconhecimento por algum tempo, pois o simples congestionamento da rede também pode ser a causa do ACK não estar chegando.

Figura 8 - Funcionamento do ataque SYN Flood.



Fonte: Retirado do site cloudflare.com.

No entanto, em um ataque, as conexões abertas no servidor, criadas pelo cliente mal-intencionado, usam recursos no servidor e podem eventualmente exceder a sua capacidade de conexões. Assim, o servidor não pode se conectar a nenhum cliente, seja legítimo ou não, criando assim a negação de serviços do servidor a outros clientes legítimos.

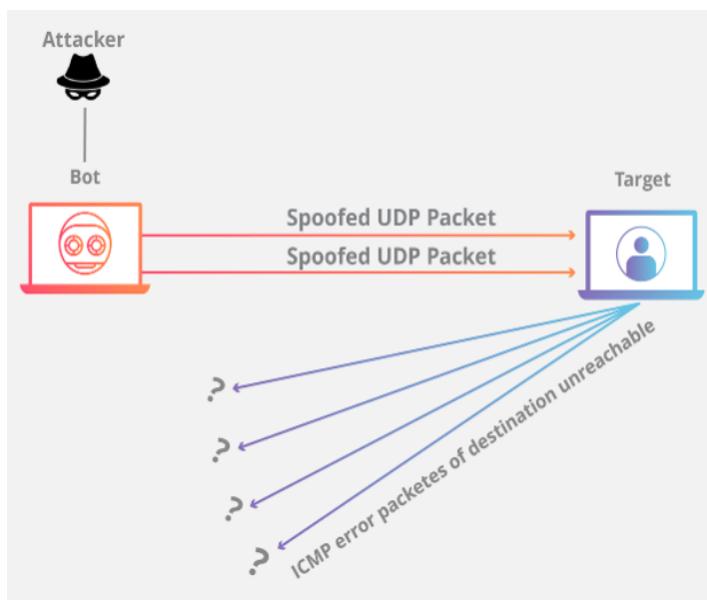
2.4.2 UDP Flood

O Ataque do tipo *UDP Flood* é um método de negação de serviço que utiliza o protocolo UDP. Diferentemente do protocolo TCP, o protocolo UDP não é orientado a conexões. Sob condições normais, quando um servidor recebe um pacote UDP ocorre o seguinte:

- O Servidor verifica primeiro se existe alguma aplicação que esteja escutando pedidos naquela porta específica.
- Caso nenhum programa esteja recebendo pacotes dessa porta, o servidor responde com um pacote ICMP para informar o remetente que o destino não está disponível.

Um ataque do tipo *UDP Flood* funciona ao enviar uma grande quantidade de pacotes UDP para portas aleatórias de um dispositivo, com um servidor. Desse modo, para um grande número de pacotes UDP, o sistema vítima será forçado a enviar muitos pacotes ICMP, o que irá deixá-lo indisponível para outros clientes, resultando na de negação de serviço. A Figura 9 apresenta o funcionamento do ataque *UDP Flood*.

Figura 9- Funcionamento do ataque UDP Flood.



Fonte: Retirado do site cloudflare.com.

2.4.3 ICMP Flood

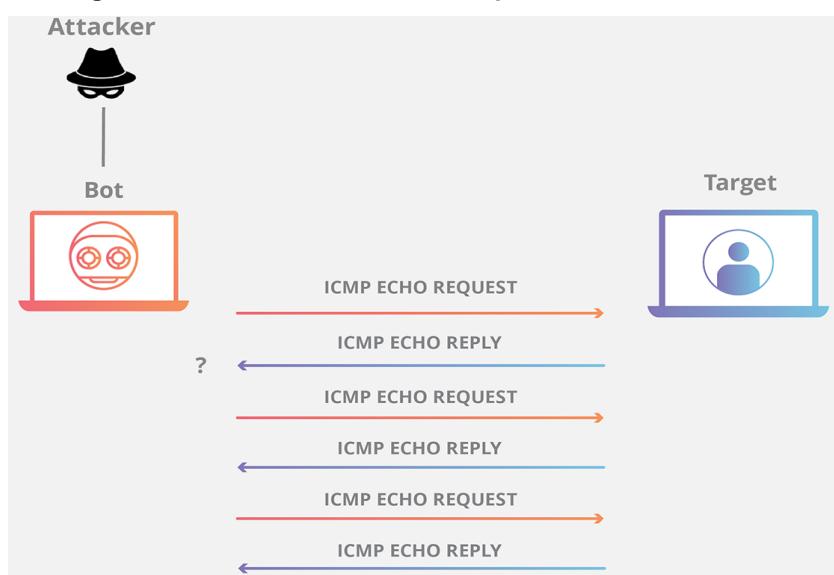
O ICMP (*Internet Control Message Protocol*), que é utilizado em um ataque ICMP *Flood*, é um protocolo de camada da Internet usado para se comunicar entre dispositivos de rede.

O ataque ICMP *Flood* é um ataque de negação de serviço em que o invasor tenta sobrecarregar um dispositivo de destino com pacotes de solicitação do protocolo ICMP, fazendo com que o destino fique inacessível ao tráfego normal. Quando o tráfego de ataque vem de vários dispositivos, o ataque se torna um ataque DDoS ou de negação de serviço distribuído.

Geralmente, as mensagens de solicitação e resposta do protocolo ICMP são usadas para executar um *ping* em um dispositivo de rede com o objetivo de diagnosticar a integridade e a conectividade do dispositivo e a conexão entre o remetente e o dispositivo.

Uma solicitação ICMP requer alguns recursos do servidor para processar cada solicitação e enviar uma resposta. A solicitação também requer largura de banda tanto na mensagem recebida (*echo-request*) quanto na resposta de saída (*echo-reply*). Desse modo, o ataque ICMP *Flood* tem como objetivo sobrecarregar a capacidade do dispositivo com tráfego falso. O funcionamento do ataque ICMP pode ser observado a partir da Figura 10.

Figura 10- Funcionamento do ataque ICMP *Flood*



Fonte: Retirado do site cloudflare.com

Conforme a Figura 10 apresenta, a forma de um ataque ICMP pode ser dividida em duas etapas que se repetem:

1. O invasor envia muitos pacotes de solicitação de ICMP ao servidor de destino usando vários dispositivos.
2. Em seguida, o servidor de destino envia um pacote de resposta de ICMP ao endereço IP de cada dispositivo solicitante como resposta.

2.4.4 Ataque Mirai

O Mirai é um *malware*, software que foi elaborado com o objetivo de causar danos a dispositivos ou sistemas, que pode controlar dispositivos IoT vulneráveis com o objetivo de criar uma rede de dispositivos infectados para conduzir ataques distribuídos de negação de serviços (DDoS) .

O ataque Mirai foi identificado primeiramente em Agosto de 2016 pelo grupo de pesquisa em segurança Malware Must Die. Esse tipo de ataque, e as suas variações, tem sido utilizado para realizar vários grandes ataques DDoS recentes, como o de Setembro de 2016 no site krebsonsecurity e em Outubro de 2016 realizado contra o provedor de serviços Dyn que retirou por algumas horas sites como Twitter, Netflix, Reddit e o GitHub.

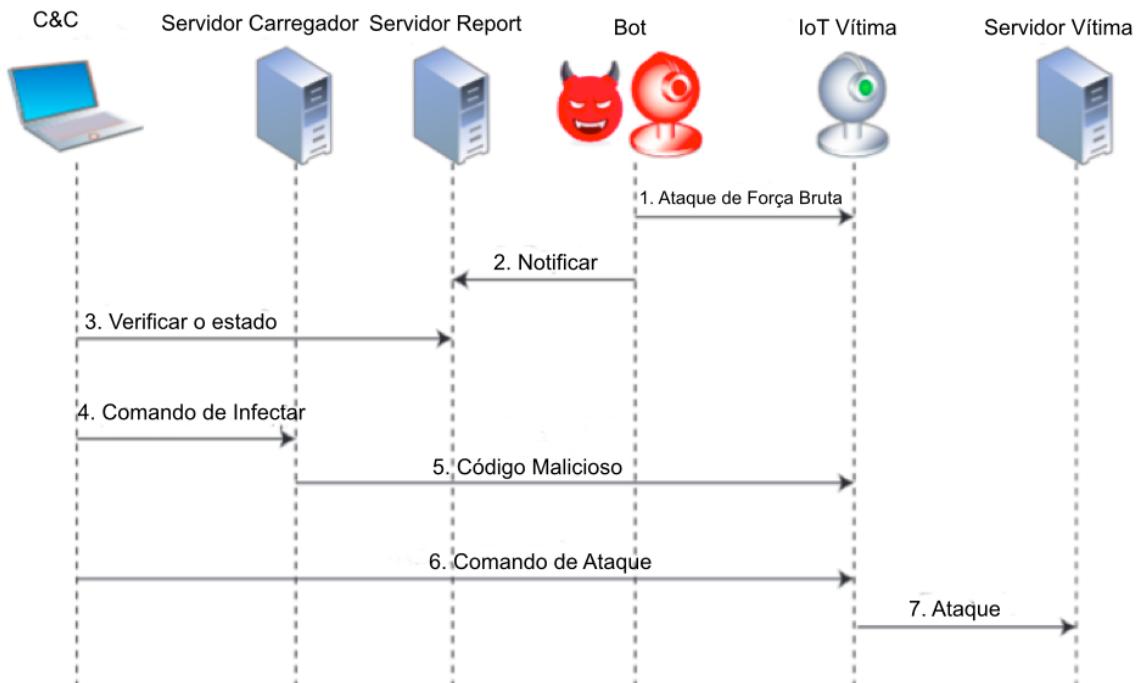
O Mirai faz a varredura de grandes partes da Internet em busca de dispositivos IoT e, em seguida, tenta efetuar acesso nesses dispositivos usando uma série de combinações de nome de usuário / senha, que são pré-configurados por padrão para vários dispositivos. Assim, o Mirai é capaz de criar uma grande rede de dispositivos infectados com esse código malicioso, as *botnets*.

Segundo KOLIAS et al (2017), e como apresentado na Figura 11, um *botnet* Mirai é composto por quatro componentes principais:

- O *bot*: é o malware que infecta dispositivos. Seu objetivo são propagar a infecção para outros dispositivos IoT , e também de realizar o ataque DoS aos dispositivos alvos.
- O centro de comando e controle (C&C): fornece ao *botmaster* uma interface de gerenciamento centralizada para verificar a condição das *botnets* e comandar novos ataques DDoS.
- The *loader* (carregador): facilita a disseminação de executáveis visando diferentes plataformas.

- Servidor *Report*: Ele mantém um banco de dados com detalhes sobre todos os dispositivos no *botnet*.

Figura 11- Funcionamento da Botnet Mirai: seus componentes e etapas.



Fonte: KOLIAS et al (2017).

Como a Figura 11 apresenta, o ataque DDoS do *malware* Mirai possui as seguintes etapas de funcionamento:

1. Inicialmente, o *bot*, dispositivo infectado com o código *malware* do Mirai, realiza um ataque de força bruta para descobrir as credenciais de outros dispositivos de IoT vulneráveis.
2. Ao descobrir as credenciais corretas e obter controle ao dispositivo, o *bot* encaminha as várias características do novo dispositivo ao *report server*.
3. Por meio do servidor C&C, o *botmaster* verifica frequentemente as novas vítimas em potencial, bem como o status atual das *botnets*, comunicando-se com o servidor de relatórios.
4. Depois de decidir quais dispositivos vulneráveis infectar, o *botmaster* emite um comando no carregador contendo todos comando necessários para o ataque a ser realizado.
5. O carregador efetua o *login* no dispositivo de destino e instrui-o a baixar e executar a versão binária correspondente do *malware*.

Assim que o *malware* é executado, ele tentará se proteger de outros *malwares*, fechando pontos de intrusão, como os serviços *telnet* e *secure shell* (SSH). Nesse ponto, a instância de *bot* recém-recrutada pode se comunicar com o servidor C&C para receber comandos de ataque.

Assim, o *botmaster* instrui todas as instâncias de *bots* a iniciarem um ataque contra um servidor de destino emitindo um comando simples através do servidor C&C com os parâmetros correspondentes, como o tipo e duração do ataque e os endereços IP das instâncias do *bot* e do servidor de destino.

6. As instâncias de *bot* começarão a atacar o servidor de destino com uma das variações de ataque disponíveis, como ataques de GRE (*Generic Routing Encapsulation*), SYN *Flood* e HTTP *Flood*.

2.5 MÉTRICAS DE DESEMPENHO

Nessa sessão serão apresentadas as métricas utilizadas para avaliar o desempenho dos algoritmos no Capítulo 4.

A matriz de confusão, também conhecida como matriz de erro, é uma tabela específica que permite a visualização do desempenho de um algoritmo de aprendizado de máquina. Cada linha da matriz representa as instâncias em uma classe prevista, enquanto cada coluna representa as instâncias em uma classe real.

Os dados que compõem a matriz de confusão são obtidos ao final do processo de aprendizado de máquina. As métricas de performance que serão apresentadas a seguir utilizam esses resultados em seus cálculos. A Tabela 1 apresenta uma matriz de confusão.

Tabela 1- Matriz de confusão.

		Valor Previsto	
		Positivo	Negativo
Valor Verdadeiro	Negativo	Verdadeiros Positivos	Falsos Negativos
	Positivo	Falsos Positivos	Verdadeiros Negativos

Fonte: VAZ (2018).

Na Tabela 1 podemos observar a matriz de confusão. No contexto dos métodos de classificação, os valores previstos são os dados atribuídos, pelo algoritmo classificador, para cada classe, enquanto que, o valor verdadeiro corresponde a real atribuição de classe para os dados.

No contexto dos experimentos do capítulo 4 os algoritmos terão a tarefa de detectar entre os dados de tráfego, aqueles dados que pertencem a um ataque, distinguindo-os dos dados que pertencem a um comportamento normal, não malicioso. Em termos da matriz de confusão, temos:

1. Verdadeiros Positivo (VP): Os dados de ataques que serão corretamente classificados como ataques.
2. Verdadeiro Negativo (VN): Os dados normais que serão corretamente classificados como normais.
3. Falso Positivo (FP): Os dados normais que serão incorretamente classificados como ataque.
4. Falso Negativo (FN): Os dados de ataque que serão incorretamente classificados como normais.

Utilizando os valores da matriz de confusão é possível obter as métricas de Acurácia, Precisão e Medida F1. A medida da acurácia representa a porcentagem de acerto de um classificador. A acurácia (A) é definida pela soma dos verdadeiros positivos (VP) e verdadeiros negativos (VN), os chamados acertos, dividido pelo todo, isto é, a soma dos falsos (FP) e verdadeiros positivos (VP) e dos falsos (FN) e verdadeiros negativos (VN). A Equação 4 apresenta a fórmula para a acurácia.

Equação 4- Fórmula da Acurácia

$$A = \frac{VP+VN}{VP+VN+FP+FN}$$

Fonte: Elaborado pelo autor.

A Precisão ou valor de predição positiva, é uma medida do quanto exato é a classificação para as amostras positivas. A precisão é definida como o número de verdadeiros positivos dividido pela soma de verdadeiros positivos mais falsos positivos, como pode-se observar na Equação 5.

Equação 5- Fórmula da Precisão

$$P = \frac{VP}{VP+FP}$$

Fonte: Elaborado pelo autor.

A medida F1 , também chamado de F1 score, é uma métrica que busca um modelo que faça um balanço entre *recall* e precisão. Ela é definida como duas vezes a média harmônica entre a revocação (R) e precisão (P), ou seja, é um meio termo entre as duas métricas anteriores. A Equação 6 abaixo apresenta a fórmula da medida F1.

Equação 6- Fórmula da Medida F1

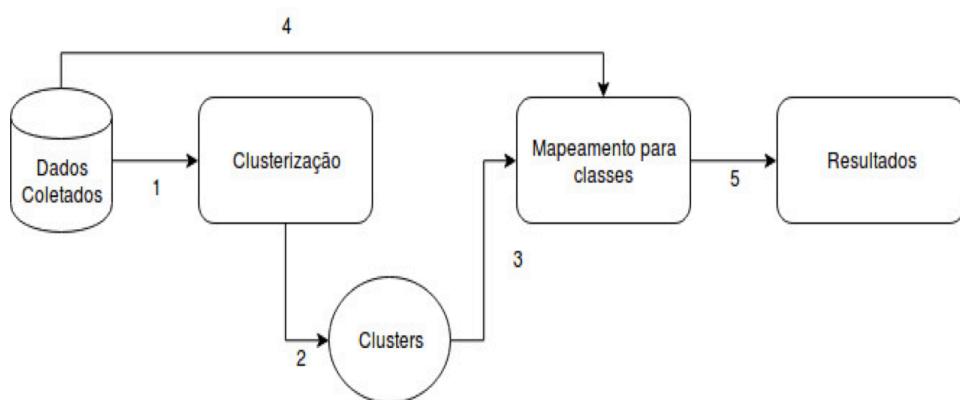
$$F1 = 2 \times \frac{P \times R}{P + R}$$

Fonte: Elaborado pelo autor.

2.6 CLASSIFICAÇÃO POR CLUSTER

A classificação por clusterização ou por cluster é uma metodologia que mescla a técnica de cluster, um método de aprendizado de máquina não supervisionado, com a técnica de classificação, um método supervisionado de aprendizado de máquina. A Figura 12 apresenta o funcionamento dessa metodologia.

Figura 12- Fluxograma da classificação via clusterização.



Fonte: LOPEZ et al. (2012), adaptado.

Essa metodologia funciona de maneira que, primeiramente, os dados da base dados irão servir de entrada para o processo de clusterização, no qual serão aplicados os algoritmos apresentados na seção 2.3, criando clusters e agrupando os dados nesses clusters. Em seguida, cada cluster será interpretado como uma classe, e servirá de entrada para o processo de classificação.

Assim, é necessário certificar-se de que o número de clusters a gerar seja o mesmo que o número de rótulos de classe no conjunto de dados para obter um modelo útil. No caso dos experimentos, teremos duas classes, a classe ataque e a classe normal. Desse modo, sempre os algoritmos de clusterização terão que necessariamente gerar dois clusters.

Primeiramente, um algoritmo de *clustering* é executado, e a base de dados é agrupada nos dois clusters. Após isso, cada cluster é associado a uma classe, um cluster é associado como a classe ataque e outro cluster é associado como a classe normal. Por fim, é realizado a verificação das métricas de desempenhos.

Uma das vantagens de se utilizar a classificação via clusterização é poder usar as métricas de desempenho comumente atribuídas para os classificadores, como as métricas apresentadas na seção 3.6.

2.7 TRABALHOS RELACIONADOS

Diferentes estudos acadêmicos têm proposto realizar a detecção ataques negação de serviço utilizando técnicas não supervisionadas, como DROMAD et al. (2017) e GHAFIR et al. (2018). Entretanto, as propostas apresentadas são baseadas em modelos com pouca presença de dispositivos IoT, os quais, como aponta JYOTHI et al. (2016), possuem comportamento muito diferente dos dispositivos mais tradicionais, como computadores e notebooks.

Além disso, muitos artigos como PERVEZ (2014), utilizam dados que resultam de simulação ou base de dados antigas, e não dados obtidos em experimentos com redes reais, o que pode distanciar os dados coletados da realidade. Como ZHONG et al. (2007), que utiliza a base de dados DARPA 1998¹, e GADDAM et al. (2007) e JIANLIANG et al. (2012) que utilizam a base de dados KDD 99², ambas bases antigas, respectivamente de 1998 e 1999, e assim não possuem dados relativos ao comportamento de tráfego de dispositivos IoT.

DOSHI et al. (2018) apresenta uma proposta de detecção de ataques de negação de serviço dentro de uma rede da Internet das Coisas. Os autores coletaram dados de tráfego de experimentos com dispositivos IoT e com ataques de negação de serviço, no caso o TCP SYN, UDP e HTTP GET, e aplicaram técnicas de Aprendizado de Máquina para detectar a ocorrência dos ataques. Nessa pesquisa, pode-se se observar o comportamento das características de tráfego de rede como atributos para o processo de AM. Entretanto, a pesquisa utiliza apenas métodos supervisionados para realizar a detecção dos ataques e não apresenta uma alternativa que utilize a metodologia não supervisionada.

Desse modo, este trabalho de conclusão de curso busca apresentar, nas próximas seções, a viabilidade da utilização da metodologia não supervisionada para a detecção de ataques na Internet das Coisas e apresentar resultados de experimentos reais que possam retratar o comportamento dos ataques e suas diferenças para o comportamento não malicioso nessas redes.

¹Disponível em: <<https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset>> Acesso em: 20 de Outubro de 2018.

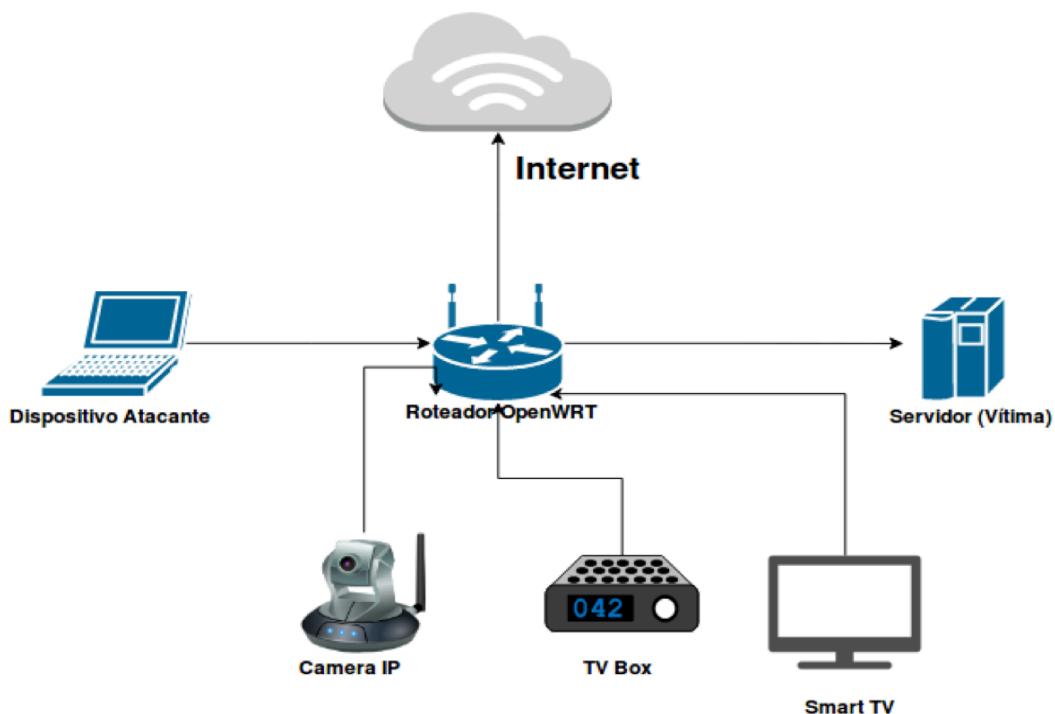
²Disponível em: <<http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>> Acesso em: 20 de Outubro de 2018.

3. METODOLOGIA – ESTUDO DE CASO

3.1 ARQUITETURA DO EXPERIMENTO

Com o objetivo de avaliar o desempenho dos algoritmos de aprendizado de máquina não supervisionado, apresentados na seção 2.3, foi realizado uma prova de conceito com base em DOSHI et. al (2018)., com respeito à detecção de ataques maliciosos em uma rede de computadores com diversos dispositivos da Internet das Coisas. A Figura 13 apresenta a arquitetura dos experimentos.

Figura 13- Arquitetura dos experimentos.



Fonte: Adaptado de DOSHI et. al (2018).

A Figura 13 apresenta a arquitetura utilizada para os experimentos realizados. Nessa figura, pode-se observar a presença de um dispositivo atacante, o qual será utilizado para realizar os ataques de negação de serviço, a presença de um servidor, que servirá como vítima dos ataques, e também, a presença dos dispositivos IoT, os quais serão utilizados como referência de tráfego normal.

Além disso, existe nessa arquitetura um roteador OpenWRT³, o qual irá coletar os dados de tráfego, através da captura dos pacotes. Todos os dispositivos estarão conectados via Wi-Fi e terão acesso à Internet.

3.2. CENÁRIOS DO EXPERIMENTO

Com base na arquitetura de experimento proposta, foram realizados dois cenários de experimentos, conforme a Tabela 2 apresenta.

Tabela 2- Descrição dos Cenários

Descrição dos Cenários	Cenário 1: KALI	Cenário 2: MIRAI
Dispositivo Atacante	1 Notebook com Kali Linux	1 Notebook com o <i>Malware Mirai</i>
Ferramenta de Ataque	Hping3	<i>Malware Mirai</i>
Dispositivo Vítima	1 Notebook com o Servidor Apache versão 2.4	1 Notebook com o Servidor Apache versão 2.4
Dispositivo Coletor de Dados	1 Roteador TP Link router WR1043ND com o OpenWRT	1 Roteador TP Link router WR1043ND com o OpenWRT
Tráfego Não Malicioso	1 <i>Smart TV</i> ; 1 Câmera IP; 1 <i>TV Box</i>	1 <i>Smart TV</i> ; 1 Câmera IP; 1 <i>TV Box</i>
Tipos de Ataques	TCP SYN Flood, UDP Flood, ICMP Flood	Mirai
Dados Coletados dos Pacotes	Protocolo, Tamanho e Tempo entre a chegada dos Pacotes	Protocolo, Tamanho e Tempo entre a chegada dos Pacotes
Tempo de Coleta para o Tráfego de Ataque	10 Minutos (10 vezes 3 ataques de 20 segundos)	10 Minutos (10 vezes 1 ataques de 60 segundos)
Tempo de Coleta para o Tráfego Normal	10 Minutos	10 Minutos
Composição da Base de Dados Gerada (%)	75% Tráfego de Ataque e 25% de Tráfego Normal	75% Tráfego de Ataque e 25% de Tráfego Normal
Composição da Base de Dados Gerada (Pacotes)	72.062 (Ataque) e 24.999 (Normal)	664.240 (Ataque) e 188.173 (Normal)

Fonte: Elaborado pelo autor.

No Cenário 1, serão realizados ataques SYN *Flood*, UDP *Flood*, ICMP *Flood*, utilizando a ferramenta Hping3⁴. A máquina atacante será um notebook com o sistema operacional Kali Linux⁵ e os ataques serão direcionados para o servidor apache versão 2.4⁶. Os dados de cada ataque serão coletados e irão compor a Base de Dados 1 (KALI).

No Cenário 2, será realizado o ataque com o *malware Mirai*. Um notebook contendo o código malicioso do Mirai será utilizado como o *bot infectado*.

³ Disponível em:< <https://openwrt.org/> > Acesso em: 20 de Outubro de 2018

⁴ Disponível em:< <https://tools.kali.org/> > Acesso em: 20 de Outubro de 2018

⁵ Disponível em:< <https://www.kali.org/> > Acesso em: 20 de Outubro de 2018.

⁶ Disponível em:< <https://httpd.apache.org/docs/2.4/pt-br/> > Acesso em: 20 de Outubro de 2018.

Os ataques são direcionados para o servidor Apache versão 2.4. Os dados de cada ataque serão coletados e irão compor a Base de Dados 2 (MIRAI).

Os dispositivos IoT foram utilizados para gerar um tráfego não malicioso, normal, que pode ser atualização de sistema operacional ou aplicativos, download de arquivos, visualização de vídeos e conteúdo web.

Conforme a Tabela 2 apresenta, foram utilizados como dispositivos IoT uma Câmera IP Foscam⁷, um TV Box BTV⁸ e uma Smart TV⁹. Para coletar o tráfego normal (não DoS), todos os três dispositivos IoT foram conectados por 10 minutos e foram coletados os arquivos de tráfego (.pcap), registrando todos os pacotes enviados durante esse período de tempo.

Para a realização dos experimentos com os métodos de aprendizagem de máquina não supervisionados, foram utilizadas as duas bases de dados geradas a partir dos cenários descritos na Tabela 2. A primeira utilizada foi criada a partir do Cenário 1. A segunda base de dados foi criada a partir do Cenário 2.

A Base de Dados 1 é composta pelos dados obtidos durante os ataques SYN Flood, UDP Flood, ICMP Flood, gerados a partir da ferramenta Hping3 do Kali Linux, e os dados obtidos durante a coleta de tráfego normal dos dispositivos IoT. Essa base de dados é composta por 75% de tráfego de ataque e 25% de tráfego normal.

A Base de Dados 2 é composta pelos dados obtidos durante o ataque Mirai e pelos dados obtidos durante a coleta de tráfego normal dos dispositivos IoT. Essa base de dados é composta por 75% de tráfego de ataque e 25% de tráfego normal.

⁷ Disponível em:<www.foscam.com.br/> Acesso em: 25 de Outubro de 2018.

⁸ Disponível em:<www.btv-box.com/> Acesso em: 25 de Outubro de 2018.

⁹ Disponível em:<www.samsung.com/smart-tv/> Acesso em: 25 de Outubro de 2018.

3.3. AVALIAÇÃO DE DESEMPENHO

Para a realização dos experimentos com os métodos não supervisionados, foram utilizados as duas base de dados geradas a partir dos dois cenários de ataques, conforme apresentado na seção 3.2. Foram coletados três características do pacote de rede, o Tamanho do Pacote, Protocolo e Tempo entre chegada dos Pacotes.

Para cada um dos cenários, todas as três características coletadas serão testadas como atributo para cada um dos métodos não supervisionado apresentados na seção 2.3. Primeiramente, serão apresentados os resultados utilizando todos os três atributos em conjunto. Em seguida, será apresentada os resultados dos algoritmos com cada característica individualmente como o atributo.

Os Métodos não supervisionados serão implementados através da ferramenta Weka¹⁰. O Weka é um conjunto de software de aprendizado de máquina escrito em Java, desenvolvido na Universidade de Waikato, Nova Zelândia. É um software livre, licenciado sob a Licença Pública GNU. O Weka contém ferramentas para preparação de dados, classificação, regressão, agrupamento, mineração de regras de associação e visualização de dados.

Os resultados obtidos por cada método e por cada característica serão avaliados de acordo com as métricas de desempenho apresentadas na seção 2.5: Acurácia, Precisão, Medida F1; e os valores da matriz de confusão, Verdadeiro Positivo, Verdadeiro Negativo, Falso Positivo e Falso Negativo.

Como nas duas bases de dados existe uma proporção de 75% de pacotes de ataque e 25% de pacotes normais, é esperado que, para que um método seja considerado minimamente efetivo, os resultados de Verdadeiro Positivo estejam acima de 75%. Assim, um método que classifique todos os pacotes como sendo pacotes de ataque deverá obter 75% de acerto para essa tarefa.

¹⁰ Disponível em:<www.cs.waikato.ac.nz/ml/weka/> Acesso em: 25 de Outubro de 2018.

4. RESULTADOS

4.1. RESULTADOS UTILIZANDO A BASE DE DADOS 1: KALI

Nesta seção serão apresentados os resultados utilizando a Base de Dados 1, a qual utilizou os dados obtidos no Cenário 1, baseado nos ataques gerados a partir da ferramenta Hping3 do Kali Linux, conforme apresentado na seção 3.2.

4.1.1 Resultados utilizando Tamanho do Pacote, Protocolo e Tempo entre chegada dos Pacotes, para a Base de Dados 1: KALI

Utilizando em conjunto o Tamanho do Pacote, Protocolo e Tempo entre chegada dos Pacotes, como atributos de entrada para o processo de aprendizado de máquina não supervisionado, foram obtidos os resultados das Tabelas 3 e 4.

Tabela 3- Resultados dos algoritmos com relação a performance das métricas acertos utilizando Tamanho do Pacote, Protocolo, Tempo entre chegada dos Pacotes para a Base de Dados 1.

MÉTODO/Performance	Acurácia	Precisão	Medida F1
K-Means (euc)	58.77%	78.40%	68.90%
K-Means (man)	66.74%	83.20%	75.60%
EM	92.77%	91.20%	95.40%
Fartherst-first	84.14%	87.10%	89.60%
DBSCAN	60.98%	80.80%	70.30%
CANOPY	59.79%	80.40%	69.10%

Fonte: Elaborado pelo autor

Tabela 4- Resultados dos algoritmos com relação aos valores relativos à matriz de confusão utilizando Tamanho do Pacote, Protocolo, Tempo entre chegada dos Pacotes para a Base de Dados 1.

MÉTODO/Performance	Verdadeiro Positivo	Falso Positivo	Verdadeiro Negativo	Falso Negativo
K-Means (euc)	44.266 (61.40%)	12.225 (48.90%)	12.774 (51.10%)	27.796 (38.60%)
K-Means (man)	49.885 (69.20%)	10.105 (40.40%)	14.894 (59.60%)	22.177 (30.80%)
EM	72.026 (99.95%)	6.979 (27.90%)	18.020 (72.10%)	36 (0.05%)
Fartherst-first	66.502 (92.30%)	9.825 (39.30%)	15.174 (60.70%)	5.560 (7.70%)
DBSCAN	44.841 (62.20%)	10.656 (42.60%)	14.343 (57.40%)	27.221 (37.80%)
CANOPY	43.676 (60.60%)	10.642 (42.60%)	14.357 (57.40%)	28.386 (39.40%)

Fonte: Elaborado pelo autor.

A Tabela 3 apresenta os resultados dos algoritmos em relação às métricas de Acurácia, Precisão e Medida F1. Nessa tabela pode-se observar que os algoritmos EM, K-Means (man), e *Fartherst-first* obtiveram valores de Medida F1 acima de 75%. Também é possível verificar que o algoritmo EM obteve o melhor resultado em termos de Acurácia, Precisão e Medida F1, com valores entre 91.20% e 95.40%. De outro modo, os algoritmos K-Means (man), DBSCAN, CANOPY, K-Means (euc), apresentaram resultados de acurácia abaixo de 75% (entre 58.77% e 66.74%).

A Tabela 4 apresenta os resultados da matriz de confusão, colocando em evidência os valores, em quantidade e porcentagem, de Verdadeiro Positivo (ataques corretamente detectados), Falso Positivo (falsos alarmes, pacotes normais classificados como pacotes de ataques), Verdadeiro Negativo (a correta classificação dos pacotes normais), Falso Negativo (ataques não detectados, ou seja, classificados como normais).

Na Tabela 4, pode-se observar que os algoritmos EM e *Fartherst-First* obtiveram os maiores valores de Verdadeiro Positivo, acima de 90% (99.95% e 92.30%, respectivamente). De outro modo, é possível observar que nenhum dos algoritmos obteve resultados de Verdadeiro Negativo acima de 75%, ou seja, nenhum dos algoritmos obteve uma detecção dos pacotes normais acima de 75%.

Além disso, pode-se observar que todos os algoritmos, com a exceção de EM, obtiveram resultados significativamente elevados de Falsos Positivos (entre 39.30% e 48.90%). Por último, é importante ressaltar que os algoritmos EM e Fartherst-first tiveram resultados de Falsos Negativos abaixo de 10% (0.05% e 7.70%, respectivamente).

Dessa forma temos que, utilizando as três características do pacote, o algoritmo EM obteve o melhor desempenho em termos das métricas apresentadas, seguido em ordem decrescente de desempenho do Fartherst-first, K-Means (man), DBSCAN, CANOPY e K-Means (euc).

4.1.2 Resultados utilizando apenas o Tamanho dos Pacotes como atributo para a Base de Dados 1.

Utilizando apenas o Tamanho dos Pacotes da Base de Dados 1 como atributo de entrada para o processo de aprendizado de máquina, foram obtidos os resultados das Tabelas 5 e 6.

Tabela 5- Resultados dos algoritmos com relação a performance das métricas acertos, utilizando apenas o Tamanho dos Pacotes como atributo, para a Base de Dados 1.

MÉTODO/Performance	Acurácia	Precisão	Medida F1
K-Means (euc)	92.61%	91.10%	95.30%
K-Means (man)	78.70%	89.32%	84.96%
EM	92.62%	91.10%	95.30%
Fartherst-first	92.60%	91.00%	95.30%
DBSCAN	92.62%	91.00%	95.30%
CANOPY	92.60%	91.10%	95.30%

Fonte: Elaborado pelo autor.

Tabela 6- Resultados dos algoritmos com relação aos valores relativos à matriz de confusão, utilizando apenas o Tamanho dos Pacotes como atributos, para a Base de Dados 1.

MÉTODO/Performance	Verdadeiro Positivo	Falso Positivo	Verdadeiro Negativo	Falso Negativo
K-Means (euc)	72.034 (99.96%)	7.145 (28.58%)	17.854 (71.42%)	28 (0.04%)
K-Means (man)	58.371 (81.00%)	6.983 (27.93%)	18.016 (72.07%)	13.691 (19.00%)
EM	72.034 (99.96%)	7.131 (28.61%)	17.868 (71.47%)	28 (0.04%)
Fartherst-first	72.034 (99.96%)	7.151 (28.60%)	17.848 (71.39%)	28 (0.04%)
DBSCAN	72.034 (99.96%)	7.134 (28.50%)	17.865 (71.50%)	28 (0.04%)
CANOPY	72.034 (99.96%)	7.145 (28.58%)	17.854 (71.42%)	28 (0.04%)

Fonte: Elaborado pelo autor.

A Tabela 5 apresenta os resultados dos algoritmos em relação às métricas de acertos. Nessa tabela pode-se observar que todos os algoritmos apresentaram valores de Medida F1 acima de 75%, sendo que apenas o K-Means (man), obteve valor diferente de 95.30%. Também é possível notar que os algoritmos obtiveram valor de Acurácia e Precisão entre 78.70% e 91.10%.

A Tabela 6 apresenta os resultados dos algoritmos com relação às taxas da matriz de confusão. Nessa tabela pode-se observar que todos os algoritmos obtiveram valores de Verdadeiro Positivo acima de 90% (com 99.96%), com exceção do K-Means (man). De outro modo, é possível observar que nenhum dos algoritmos obteve resultados de Verdadeiro Negativo acima de 75%.

Além disso, a Tabela 6 apresenta os resultados em termos de Falso Positivo e Falso Negativo. Pode-se observar que os algoritmos obtiveram resultados de Falso Positivo entre 27.93% e 28.61%. Por último, é importante ressaltar que todos os algoritmos, com exceção do K-Means (man), tiveram resultados de Falso Negativo abaixo de 10% (entre 0.05% e 7.70%).

Dessa forma temos que utilizando apenas o Tamanho do Pacote como atributo, todos os algoritmos obtiveram resultados similares em termos das métricas apresentadas, com exceção do K-Means (man), que obteve desempenho inferior aos demais métodos.

4.1.3 Resultados utilizando apenas o Protocolo como Atributo, para a Base de Dados 1: KALI.

Utilizando apenas o Protocolo como atributo de entrada para o processo de aprendizado de máquina, foram obtidos os resultados das Tabelas 7 e 8.

Tabela 7- Resultados dos algoritmos com relação a performance das métricas acertos, utilizando apenas o Protocolo como atributo, para a Base de Dados 1.

MÉTODO/Performance	Acurácia	Precisão	Medida F1
K-Means (euc)	57.72%	76.10%	68.80%
K-Means (man)	57.72%	76.10%	68.80%
EM	60.10%	79.60%	69.90%
Fartherst-first	57.32%	74.60%	69.20%
DBSCAN	57.72%	76.10%	68.80%
CANOPY	42.83%	72.60%	59.80%

Fonte: Elaborado pelo autor.

Tabela 8- Resultados dos algoritmos com relação à matriz de confusão, utilizando apenas o Protocolo como atributo, para Base de Dados 1.

MÉTODO/Performance	Verdadeiro Positivo	Falso Positivo	Verdadeiro Negativo	Falso Negativo
K-Means (euc)	45.261 (62.80%)	14.231 (56.90%)	10.768 (43.10%)	26.801 (37.20%)
K-Means (man)	45.261 (62.80%)	14.231 (56.90%)	10.768 (43.10%)	26.801 (37.20%)
EM	44.871 (62.30%)	11.533 (46.10%)	13.466 (53.90%)	27.191 (37.70%)
Fartherst-first	46.436 (64.40%)	15.791 (63.20%)	9.208 (36.80%)	25626 (35.60%)
DBSCAN	45.261 (62.80%)	14.231 (56.90%)	10.768 (43.10%)	26.801 (37.20%)
CANOPY	28.108 (50.80%)	10.618 (44.10%)	13.468 (55.90%)	27.194 (49.20%)

Fonte: Elaborado pelo autor.

A Tabela 7 apresenta os resultados dos algoritmos em relação às métricas de acertos. Nessa tabela pode-se observar que todos os algoritmos apresentaram valores de Medida F1 menores que 75%. Também é possível notar que os algoritmos obtiveram valor de Acurácia e Precisão entre 42.83% e 79.60%.

A Tabela 8 apresenta os resultados dos algoritmos com relação à matriz de confusão. Nessa tabela pode-se observar que todos os algoritmos obtiveram valores de Verdadeiro Positivo abaixo de 75%. Além disso, é possível observar que nenhum dos algoritmos obteve resultados de Verdadeiro Negativo acima de 75%.

Além disso, a Tabela 8 apresenta os resultados em termos de Falso Positivo e Falso Negativo. Pode-se observar que todos os algoritmos, obtiveram resultados de Falsos Positivos significativamente elevados (entre 44.10% e 63.20%). Por último, é importante ressaltar que todos os algoritmos tiveram resultados de Falsos Negativos muito acima de 10% (entre 35.60% e 49.20%).

Dessa forma temos que utilizando apenas o Protocolo como atributo, todos os algoritmos obtiveram resultados muito próximos em termos das métricas apresentadas, com a exceção do CANOPY, que obteve desempenho inferior aos demais métodos.

4.1.4. Resultados utilizando apenas o Tempo entre chegada dos Pacotes como atributo, para a Base de Dados 1 (KALI) .

Utilizando apenas o Tempo entre chegada dos Pacotes da Base de Dados 1 como atributo de entrada para o processo de aprendizado de máquina, foram obtidos os resultados das Tabelas 9 e 10.

Tabela 9- Resultados dos algoritmos com relação a performance das métricas acertos, utilizando apenas o Tempo entre chegada dos Pacotes como atributo para a Base de Dados 1

MÉTODO/Performance	Acurácia	Precisão	Medida F1
K-Means (euc)	89.10%	87.90%	93.10%
K-Means (man)	91.17%	97.80%	93.80%
EM	94.90%	98.70%	96.50%
Fartherst-first	75.65%	75.34%	85.90%
DBSCAN	88.92%	87.70%	93.00%
CANOPY	85.09%	85.20%	91.70%

Fonte: Elaborado pelo autor.

Tabela 10- Resultados dos algoritmos com relação à matriz de confusão, utilizando apenas o Tempo entre chegada dos Pacotes como atributo para a Base de Dados 1.

MÉTODO/Performance	Verdadeiro Positivo	Falso Positivo	Verdadeiro Negativo	Falso Negativo
K-Means (euc)	71.277 (98.90%)	9.794 (39.20%)	15.205 (60.80%)	785 (1.10%)
K-Means (man)	64.980 (90.20%)	1.479 (5.90%)	23.520 (94.10%)	7.082 (9.80%)
EM	68.016 (94.40%)	864 (3.50%)	24.135 (96.50%)	4.046 (5.60%)
Fartherst-first	71.995 (99.91%)	23.565 (94.30%)	1.434 (5.70%)	67 (0.09%)
DBSCAN	71.308 (99.00%)	9.998 (40.00%)	15.001 (60.00%)	754 (1.0%)
CANOPY	71.570 (99.30%)	12.462 (53.10)	11.015 (46.90%)	492 (0.7%)

Fonte: Elaborado pelo autor.

A Tabela 9 apresenta os resultados dos algoritmos em relação às métricas de acertos. Nessa tabela pode-se observar que todos os algoritmos apresentaram valores de Medida F1 superiores à 75% (entre 85.90% e 93.80%). Também é possível notar que os algoritmos obtiveram valor de Acurácia e Precisão entre 75.34% e 98.70%.

A Tabela 10 apresenta os resultados dos algoritmos com relação à matriz de confusão. Nessa tabela pode-se observar que todos os algoritmos obtiveram valores de valores de Verdadeiro Positivo acima de 90% (entre 90.20% e 99.91%). Além disso, é possível observar que os algoritmos EM e K-Means (man) e obtiveram

resultados de Verdadeiro Negativo acima de 90%, ou seja, uma detecção dos pacotes normais acima de 90%. Por outro lado, os demais métodos tiveram resultados de Verdadeiro Negativo abaixo de 75%.

Além disso, a Tabela 10 apresenta os resultados em termos de Falso Positivo e Falso Negativo. Pode-se observar que os algoritmos EM e K-Means (man) apresentaram valores de Falso Positivo menores que 10% (entre 3.50% e 5.90%). De outro modo, os demais algoritmos apresentaram valores significativamente elevados de Falsos Positivos (entre 39.20% e 94.30%). Por último, é importante ressaltar que todos os algoritmos tiveram resultados de falsos negativos menores de 10% (entre 0.09% e 9.80%).

Dessa forma temos que utilizando apenas o Tempo entre a chegada dos Pacotes como atributo, o algoritmo EM obteve o melhor desempenho em termos das métricas apresentadas, seguido em ordem decrescente de desempenho o K-Means (man), K-Means (euc), DBSCAN, CANOPY e *Fartherst-first*.

4.1.5. Conclusão dos Resultados para Base de Dados 1 (KALI).

Os resultados apresentados na seção 4.1 mostram o desempenho que a metodologia não supervisionada e os algoritmos utilizados tiveram na detecção dos ataques do Cenário 1, do tipo *SYN Flood*, *UDP Flood* e *ICMP Flood*. A Tabela 11 apresenta um resumo comparativo dos resultados para Base de Dados 1.

Tabela 11- Resumo Comparativo dos Resultados para a Base de Dados 1 (KALI).

Resultados	Medida F1	Verdadeiro Positivo (VP) e Verdadeiro Negativo (VN)	Falso Positivo (FP) e Falso Negativo (FN)	Melhores Resultados	Tabelas Correspondentes
KALI: Protocolo, Tamanho e Tempo entre Pacotes	Entre 68.90% e 95.40%	<p>VP: Entre 61.40% e 99.95%. Apenas EM e Fartherst-First acima de 90%.</p> <p>VN: Entre 51.10% e 72.10%. Nenhum dos algoritmos obteve resultados de Verdadeiro Negativo acima de 75%.</p>	<p>FP: Entre 27.90% e 48.90%. Nenhum dos algoritmos obteve resultados abaixo de 10%.</p> <p>FN: Entre 0.05% e 39.40%. Apenas EM e Fartherst-First tiveram resultados abaixo de 10%.</p>	EM : F1 95.40% VP 99.95% VN 72.10% FP 27.90% FN 0.05%	Tabelas 3 e 4
KALI: Tamanho do Pacote	Entre 84.96% e 95.30%	<p>VP: Entre 81.00% e 99.96%. Todos, com exceção do K-Means (man), acima de 90%.</p> <p>VN: Entre 71.39% e 72.07%. Nenhum dos algoritmos obteve resultado acima de 75%.</p>	<p>FP: Entre 27.93% e 28.61%. Nenhum, dos algoritmos obteve resultados abaixo de 10%.</p> <p>FN: Entre 0.04% e 19.00%. Todos os métodos, com exceção do K-Means (man), tiveram resultados abaixo de 10%.</p>	EM, Fartherst-First, DBSCAN, CANOPY, K-Means(euc) obtiveram resultados similares: F1 95.30% VP 99.96% VN 71.42% FP 28.58% FN 0.04%	Tabelas 5 e 6
KALI: Protocolo	Entre 59.80% e 69.90%	<p>VP: Entre 50.80% e 64.40%. Nenhum dos algoritmos obteve resultado acima de 75%.</p> <p>VN: Entre 36.80% e 56.90%. Nenhum dos algoritmos obteve resultados de Verdadeiro Negativo acima de 75%.</p>	<p>FP: Entre 44.10% e 53.90%. Nenhum, dos algoritmos obteve resultados abaixo de 10%.</p> <p>FN: Entre 35.60% e 49.20%. Nenhum, dos algoritmos obteve resultado abaixo de 10%.</p>	EM : F1 69.90% VP 62.30% VN 53.90% FP 46.10% FN 37.70%	Tabelas 7 e 8
KALI: Tempo entre Pacotes	Entre 85.90% e 93.90%	<p>VP: Entre 90.20% e 99.91%. Todos acima de 90%.</p> <p>VN: Entre 5.70% e 96.50%. EM e K-Means (man) e obtiveram resultado o acima de 90%.</p>	<p>FP: Entre 3.50% e 94.30%. EM e K-Means (man) obtiveram resultado abaixo de 10%.</p> <p>FN: Entre 0.09% e 9.00%. Todos os métodos tiveram resultados abaixo de 10%.</p>	EM: F1 96.50% VP 94.40% VN 96.50% FP 3.50% FN 5.60%	Tabelas 9 e 10

Fonte: Elaborado pelo Autor.

Na Tabela 11 pode-se observar o comportamento dos algoritmos de acordo com o uso do de cada característica (Tamanho, Protocolo e Tempo entre Pacotes) como atributo e o uso das três características em conjunto.

Em termos de Medida F1, temos que a utilização do Tamanho do Pacote e do Tempo entre chegada dos Pacotes obtiveram os melhores resultados, já que com essas características, a maioria dos métodos teve resultado acima de 90%. A utilização de todas as três características de rede em conjunto teve, para a maioria dos algoritmos, desempenho inferior a 90% em termos da Medida F1, e o uso do Protocolo teve o pior desempenho dentre as características do pacote de rede.

Quanto à detecção correta dos pacotes, ou seja, em termos de Verdadeiro Positivo (pacotes de ataque) e Verdadeiro Negativo (pacotes normais), temos que, os algoritmos conseguiram atingir valores de VP acima de 90% com a utilização das três características de tráfego em conjunto, com a utilização do Tamanho do Pacote e também com a utilização do Tempo entre chegada dos Pacotes. Por outro lado, temos que apenas com a utilização do Tempo entre a chegada dos Pacotes foi possível obter, ao mesmo tempo, valores de VP e VN acima de 90%, o que é um resultado significativamente positivo.

Em termos da classificação incorreta dos pacotes, ou seja, em termos de Falso Positivo (falsos alarmes, pacotes normais classificados como pacotes de ataque) e Falso Negativo (pacotes de ataque classificados como pacotes normais), temos que, apenas com a utilização do EM e K-Means (man) com o Tempo entre chegada dos pacotes foi possível obter valores de FN abaixo de 10%. Além disso, pode-se observar que apenas com o uso do protocolo como atributo que não houve algoritmos conseguisse obter valores de FN abaixo de 10%.

4.2. RESULTADOS UTILIZANDO A BASE DE DADOS 2

Nesta seção serão apresentados os resultados utilizando a Base de Dados 2, a qual utilizou os dados obtidos no Cenário 2, baseado nos ataques gerados a partir do ataque DDoS Mirai, conforme apresentado na seção 3.2. Primeiramente, serão apresentados os resultados utilizando todos três atributos tamanho do pacote, protocolo e tempo entre chegada dos pacotes. Em seguida, será apresentada os resultados dos algoritmos com cada uma das características como o único atributo.

4.2.1 Resultados utilizando o Protocolo, Tamanho dos Pacotes e Tempo entre chegada dos Pacotes como atributos, para a Base de Dados 2.

Utilizando o Protocolo, Tamanho dos Pacotes e o Tempo entre chegada dos Pacotes, como atributos para o processo de Aprendizado de Máquina foram obtidos os resultados das Tabelas 12 e 13.

Tabela 12- Resultados dos algoritmos com relação a performance das métricas acertos, utilizando o Protocolo, Tamanho dos Pacotes e o Tempo entre Chegada dos Pacotes como atributos para a Base de Dados 2.

MÉTODO/Performance	Acurácia	Precisão	Medida F1
K-Means (euc)	66.90%	82.40%	77.50%
K-Means (man)	66.92%	82.50%	77.50%
EM	72.83%	99.84%	78.91%
Fartherst-first	78.54%	86.00%	86.30%
DBSCAN	67.11%	82.70%	77.60%
CANOPY	66.91%	82.45%	77.49%

Fonte: Elaborado pelo autor.

Tabela 13- Resultados dos algoritmos com relação à matriz de confusão, utilizando o Protocolo, Tamanho dos Pacotes e o Tempo entre chegada dos Pacotes como atributos para a Base de Dados 2.

MÉTODO/Performance	Verdadeiro Positivo	Falso Positivo	Verdadeiro Negativo	Falso Negativo
K-Means (euc)	485.520 (73.10%)	103.425 (55.00%)	84.748 (45.00%)	178.720 (26.90%)
K-Means (man)	485.519 (73.10%)	103.210 (54.80%)	84.963 (45.20%)	178.721 (26.90%)
EM	433.299 (65.23%)	700 (0.37%)	187.471 (99.63%)	230.941 (34.80%)
Fartherst-first	575.047 (86.60%)	93.730 (49.80%)	94.443 (50.20%)	89.193 (13.40%)
DBSCAN	485.593 (73.10%)	101.635 (54.00%)	86.538 (46.00%)	178.647 (26.90%)
CANOPY	485.520 (73.09%)	103.339 (54.92%)	84.834 (45.08%)	178.72 (26.91%)

Fonte: Elaborado pelo autor.

A Tabela 12 apresenta os resultados dos algoritmos em relação às métricas de acertos. Nessa tabela pode-se observar que todos os algoritmos obtiveram valores de Medida F1 acima de 75%. Também é possível notar que os métodos EM e *Fartherst-first* tiveram os melhores resultados, com Acurácia e Precisão entre 72.83% e 99.84%.

A Tabela 13 apresenta os resultados dos algoritmos com relação à matriz de confusão. Nessa tabela pode-se observar que somente o algoritmo *Fartherst-first* obteve valor de Verdadeiro Positivo acima de 75%, com 86.60%. De outro modo, é possível observar que apenas o algoritmo EM obteve resultados de Verdadeiro Negativo acima de 75%, com 99.63%.

Além disso, a Tabela 13 apresenta os resultados em termos de Falso Positivo e Falso Negativo. Pode-se observar que todos os algoritmos, com a exceção de EM, obtiveram resultados significativamente altos de Falsos Positivos (entre 49.80% e 55.00%). Por último, é importante ressaltar que os nenhum dos algoritmos teve resultados de Falsos Negativos abaixo de 10%.

Dessa forma temos que utilizando as três características do pacote(Protocolo, Tamanho e tempo entre a chegada dos pacotes), o algoritmo *Fartherst-first* obteve o melhor desempenho em termos das métricas apresentadas, seguido em ordem decrescente de desempenho do, EM, DBSCAN, K-Means (man), K-Means (euc) e CANOPY.

4.2.2 Resultados utilizando apenas o Tamanho dos Pacotes como atributo, para a Base de Dados 2.

Utilizando apenas o Tamanho dos Pacotes como atributo para o processo de aprendizado de máquina foram obtidos os resultados das Tabelas 14 e 15.

Tabela 14- Resultados dos algoritmos com relação a performance das métricas acertos, utilizando apenas o Tamanho dos Pacotes como atributo para a Base de Dados 2.

MÉTODO/Performance	Acurácia	Precisão	Medida F1
K-Means (euc)	77.67%	90.80%	84.70%
K-Means (man)	77.69%	90.80%	84.70%
EM	93.09%	92.45%	95.73%
Fartherst-first	93.68%	92.50%	96.10%
DBSCAN	77.74%	90.90%	84.80%
CANOPY	91.16%	92.26%	94.46%

Fonte: Elaborado pelo autor.

Tabela 15- Resultados dos algoritmos com relação à matriz de confusão, utilizando apenas o tamanho dos pacotes como atributo para a base de dados 2.

MÉTODO/Performance	Verdadeiro Positivo	Falso Positivo	Verdadeiro Negativo	Falso Negativo
K-Means (euc)	527.524 (79.40%)	53.620 (28.50%)	134.553 (71.50%)	136.716 (20.60%)
K-Means (man)	527.523 (79.40%)	53.387 (28.40%)	134.786 (71.60%)	136.717 (20.60%)
EM	659.240 (99.25%)	53.387 (28.40%)	134.786 (71.60%)	5.000 (0.75%)
Fartherst-first	664.147 (99.99%)	53.870 (28.60%)	134.303 (71.40%)	93 (0.01%)
DBSCAN	527.504 (79.40%)	52.987 (28.20%)	135.186 (71.80%)	136.736 (20.60%)
CANOPY	642.804 (96.77%)	53.905 (28.65%)	134.268 (71.35%)	21.436 (3.23%)

Fonte: Elaborado pelo autor.

A Tabela 14 apresenta os resultados dos algoritmos em relação às métricas de acertos. Nessa tabela pode-se observar que todos os algoritmos apresentaram valores de Medida F1 acima de 75%, sendo que o algoritmo Fartherst-first, obteve o maior valor com 96.10%. Também é possível notar que os algoritmos avaliados obtiveram valor de acurácia e precisão entre, 77.67% e 93.68%.

A Tabela 15 apresenta os resultados dos algoritmos com relação à matriz de confusão. Nessa tabela pode-se observar que os algoritmos EM, Fartherst-first e CANOPY, obtiveram valores de Verdadeiro Positivo, acima de 90% (entre 99.25%

e 96.77%. De outro modo, é possível observar que nenhum dos algoritmos obteve resultados de Verdadeiro Negativo acima de 75%, ou seja, nenhum dos algoritmos obteve uma detecção dos pacotes normais acima de 75%.

Além disso, a Tabela 15 apresenta os resultados em termos de falso positivo e falso negativo. Pode-se observar que todos os algoritmos, com a obtiveram resultados de falsos positivos (entre 28.20% e 28.65%). Por último, é importante ressaltar que os algoritmos EM, Fartherst-first e CANOPY tiveram resultados de falsos negativos abaixo de 10% (entre 0.01% e 3.23%).

Dessa forma temos que utilizando apenas o Tamanho pacotes como atributo, o algoritmo *Fartherst-first* obteve o melhor desempenho em termos das métricas apresentadas, seguido em ordem decrescente de desempenho do EM, CANOPY, K-Means (man), DBSCAN, e K-Means (euc).

4.2.3 Resultados utilizando apenas o Protocolo como atributo, para a Base de Dados 2.

Utilizando apenas o protocolo como atributo para o processo de aprendizado de máquina, foram obtidos os resultados das Tabelas 16 e 17.

Tabela 16- Resultados dos algoritmos com relação a performance das métricas acertos, utilizando apenas o Protocolo como atributo para a Base de Dados 2.

MÉTODO/Performance	Acurácia	Precisão	Medida F1
K-Means (euc)	67.71%	81.40%	78.60%
K-Means (man)	66.55%	82.00%	77.30%
EM	66.68%	82.10%	77.40%
Fartherst-first	67.75%	81.50%	78.60%
DBSCAN	67.72%	81.40%	78.60%
CANOPY	66.31%	81.70%	77.20%

Fonte: Elaborado pelo autor.

Tabela 17- Resultados dos algoritmos com relação à matriz de confusão, utilizando apenas o Protocolo como atributo para a Base de dados 2.

MÉTODO/Performance	Verdadeiro Positivo	Falso Positivo	Verdadeiro Negativo	Falso Negativo
K-Means (euc)	503.927 (75.90%)	114.894 (61.10%)	73.279 (38.90%)	160.313 (24.10%)
K-Means (man)	485.903 (73.20%)	106.780 (56.70%)	81.393 (43.30%)	178.337 (26.80%)
EM	485.860 (73.10%)	105.607 (56.10%)	82.560 (43.90%)	178.380 (26.90%)
Fartherst-first	503.543 (75.80%)	114.181 (60.70%)	73.992 (39.30%)	160.695 (24.20%)
DBSCAN	503.956 (75.90%)	114.894 (61.10%)	73.279 (38.90%)	160.284 (24.10%)
CANOPY	486.110 (73.20%)	109.036 (57.90%)	79.137 (42.10%)	178.130 (26.80%)

Fonte: Elaborado pelo autor.

A Tabela 16 apresenta os resultados dos algoritmos em relação às métricas de acertos. Nessa tabela pode-se observar que todos os algoritmos apresentaram valores de Medida F1 maiores que 75%. Também é possível notar que os algoritmos obtiveram valor de acurácia e precisão entre, 66.31% e 82.10%.

A Tabela 17 apresenta os resultados dos algoritmos com relação à matriz de confusão. Nessa tabela pode-se observar que os algoritmos K-Means (euc), DBSCAN, *Fartherst-first* obtiveram valores de verdadeiro positivo acima de 75%. Além disso, é possível observar que nenhum dos algoritmos obteve resultados de verdadeiro negativo acima de 75%, com valores entre 38.90% e 43.90%.

Além disso, a Tabela 17 apresenta os resultados em termos de falso positivo e falso negativo. Pode-se observar que todos os algoritmos apresentaram valores de falso positivo muito superiores a 10% (entre 56.10% e 61.10%). Por último, é importante ressaltar que todos os algoritmos tiveram resultados de falsos negativos superiores a 10% (entre 24.10% e 26.90%).

Dessa forma temos que utilizando apenas o Protocolo como atributo, o algoritmo *Fartherst-first* e DBSCAN, K-Means (euc) obtiveram os melhores desempenhos em termos das métricas apresentadas, seguido em ordem decrescente de desempenho do EM, K-Means (man), CANOPY.

4.2.4 Resultados utilizando apenas o Tempo entre chegada dos Pacotes como atributo, usando a Base de Dados 2.

Utilizando apenas o tempo entre chegada dos pacotes como atributo para o processo de aprendizado de máquina foram obtidos os resultados das Tabelas 18 e 19.

Tabela 18- Resultados dos algoritmos com relação a performance das métricas acertos, utilizando apenas o Tempo entre chegada dos Pacotes como atributo para a Base de Dados 2.

MÉTODO/Performance	Acurácia	Precisão	Medida F1
K-Means (euc)	99.15%	99.04%	99.46%
K-Means (man)	99.18%	99.11%	99.48%
EM	95.35%	99.98%	96.93%
Fartherst-first	79.03%	78.83%	88.14%
DBSCAN	98.79%	99.93%	99.22%
CANOPY	94.81%	93.85%	96.77%

Fonte: Elaborado pelo autor.

Tabela 19- Resultados dos algoritmos com relação à matriz de confusão, utilizando apenas o Tempo entre chegada dos Pacotes como atributo para a Base de Dados 2.

MÉTODO/Performance	Verdadeiro Positivo	Falso Positivo	Verdadeiro Negativo	Falso Negativo
K-Means (euc)	663.406 (99.87%)	6.418 (3.41%)	181.755 (96.59%)	834 (0.13%)
K-Means (man)	663.212 (99.85%)	5.969 (3.17%)	182.204 (96.83%)	1.028 (0.15%)
EM	624.651 (94.04%)	30 (0.02%)	188.143 (99.98%)	39.589 (5.96%)
Fartherst-first	663.777 (99.93%)	178.254 (94.73%)	9.919 (5.27%)	463 (0.07%)
DBSCAN	654.399 (98.52%)	456 (0.24%)	187.717 (99.76%)	9841 (1.48%)
CANOPY	663.461 (99.88%)	43.443 (23.09%)	144.730 (76.91%)	779 (0.12%)

Fonte: Elaborado pelo autor.

A Tabela 18 apresenta os resultados dos algoritmos em relação às métricas de acertos. Nessa tabela pode-se observar que todos os algoritmos apresentaram valores de Medida F1 superiores à 75% (entre 88.14% e 99.48%). Também é possível notar que os algoritmos obtiveram valor de Acurácia e Precisão entre, 78.83% e 99.98%.

A Tabela 19 apresenta os resultados dos algoritmos com relação à matriz de confusão. Nessa tabela pode-se observar que todos os algoritmos obtiveram valores de valores de Verdadeiro Positivo acima de 90% (entre 94.04% e 99.93%). Além disso, é possível observar que todos os obtiveram resultados de Verdadeiro Negativo acima de 90%, com exceção do *Fartherst-first* e CANOPY.

Além disso, a Tabela 19 apresenta os resultados em termos de Falso Positivo e Falso Negativo. Pode-se observar que todos os algoritmos, com exceção do *Fartherst-first* e CANOPY, apresentam valores de Falso Positivo menores que 10% (entre 0.02% e 3.41%). Por último, é importante ressaltar que todos os algoritmos tiveram resultados de Falsos Negativos menores de 10% (entre 0.07% e 5.96%).

Dessa forma temos que utilizando apenas o Tempo entre a chegada dos Pacotes como atributo, o algoritmo K-Means (man), obteve o melhor desempenho em termos das métricas apresentadas, seguido em ordem decrescente de desempenho o K-Means (euc), DBSCAN, EM, CANOPY e *Fartherst-first*.

4.2.5 Conclusão dos Resultados para Base de Dados 2 (MIRAI)

Os resultados apresentados na seção 4.2 mostram o desempenho que a metodologia não supervisionada e os algoritmos utilizados tiveram na detecção dos ataques do Cenário 2, do tipo Mirai. A Tabela 20 apresenta um resumo comparativo dos resultados para Base de Dados 2.

Tabela 20- Resumo Comparativo dos Resultados para a Base de Dados 2 (MIRAI).

Resultados	Medida F1	Verdadeiro Positivo (VP) e Verdadeiro Negativo (VN)	Falso Positivo (FP) e Falso Negativo (FN)	Melhores Resultados	Tabelas Correspondentes
MIRAI: Protocolo, Tamanho e Tempo entre Pacotes	Entre 77.49% e 86.30%	VP: Entre 65.23% e 86.60%. Apenas o Fartherst-First obteve resultado acima de 75%. VN: Entre 45.00% e 99.63%. Apenas o EM obteve resultado acima de 75%.	FP: Entre 0.37% e 55.00%. Apenas o EM obteve resultado abaixo de 10%. FN: Entre 13.40% e 34.80%. Nenhum dos algoritmos tiveram resultados abaixo de 10%.	Fartherst-First: F1: 86.30% VP: 86.60% VN: 50.20% FP: 49.80% FN: 13.40%	Tabelas 12 e 13
MIRAI: Tamanho do Pacote	Entre 84.70% e 96.10%	VP: Entre 79.40% e 99.99%. Fartherst-First, EM e CANOPY acima de 90%. VN: Entre 71.35% e 71.80%. Nenhum dos algoritmos obteve resultado acima de 75%.	FP: Entre 28.20% e 28.65%. Nenhum dos algoritmos obteve resultados abaixo de 10%. FN: Entre 0.01% e 20.60%. Fartherst-First, EM e CANOPY tiveram resultados abaixo de 10%.	Fartherst-First: F1: 96.10% VP: 99.99% VN: 71.40% FP: 28.60% FN: 0.01%	Tabelas 14 e 15
MIRAI: Protocolo	Entre 77.20% e 78.60%	VP: Entre 73.10% e 75.90%. K-Means (euc), DBSCAN e Fartherst-First obtiveram resultado acima de 75%. VN: Entre 38.90% e 43.90%. Nenhum dos algoritmos obteve resultados de Verdadeiro Negativo acima de 75%.	FP: Entre 56.10% e 61.10%. Nenhum dos algoritmos obteve resultados abaixo de 10%. FN: Entre 24.10% e 26.90%. Nenhum dos algoritmos obteve resultado abaixo de 10%.	K-Means (euc), DBSCAN e Fartherst-First obtiveram resultados similares: F1: 78.60% VP: 75.90% VN: 38.90% FP: 61.10% FN: 24.10%	Tabelas 16 e 17
MIRAI: Tempo entre Pacotes	Entre 88.14% e 99.48%	VP: Entre 94.04% e 99.93%. Todos acima de 90%. VN: Entre 5.27% e 99.98%. Todos, com exceção do Fartherst-First e CANOPY, obtiveram resultado o acima de 90%.	FP: Entre 0.02% e 94.73%. Todos, com exceção do Fartherst-First e CANOPY, obtiveram resultado abaixo de 10%. FN: Entre 0.07% e 5.96%. Todos os métodos tiveram resultados abaixo de 10%.	K-Means (man): F1: 99.48% VP: 99.85% VN: 96.83% FP: 3.17% FN: 0.15%	Tabelas 18 e 19

Fonte: Elaborado pelo Autor.

Na Tabela 20 pode-se observar o comportamento dos algoritmos de acordo com o uso de cada característica (Tamanho, Protocolo e Tempo entre Pacotes) como atributo e o uso das três características em conjunto.

Em termos de Medida F1, temos que a utilização do Tempo entre chegada dos Pacotes e do Tamanho do Pacote obtiveram os melhores resultados, já que com essas características, a maioria dos métodos teve resultado acima de 90%.

A utilização de todas as três características de rede em conjunto teve, desempenho inferior em termos da Medida F1, seguido do uso do Protocolo, o qual teve o pior desempenho dentre as características do pacote de rede em termos da Medida F1.

Quanto a de detecção correta dos pacotes, ou seja, em termos de Verdadeiro Positivo (pacotes de ataque) e Verdadeiro Negativo (pacotes normais), temos que os algoritmos conseguiram atingir valores de VP acima de 90% com utilização do Tamanho do Pacote e com a utilização do Tempo entre chegada dos Pacotes. Por outro lado, temos que apenas com a utilização do Tempo entre a chegada dos Pacotes, foi possível obter ao mesmo tempo valores de VP e VN acima de 90%, o que é um resultado significativamente positivo.

Em termos da classificação incorreta dos pacotes, ou seja, em termos de Falso Positivo (falsos alarmes, pacotes normais classificados como pacotes de ataque) e Falso Negativo (pacotes de ataque classificados como pacotes normais), temos que foi possível obter valores de FP abaixo de 10%, com a utilização do EM com as três características do pacote de rede, e com K-Means(euc), K-Means(man), EM, DBSCAN utilizando o Tempo entre a chegada dos Pacotes. Além disso, pode-se observar que só foi possível obter ambos valores de FP e FN menores que 10% com a utilização do Tempo entre chegada dos Pacotes.

4.3 CONCLUSÃO DOS RESULTADOS

As Seções 4.1 E 4.2 apresentaram os resultados que a metodologia não supervisionada e os algoritmos utilizados tiveram na detecção dos ataques na Base de Dados 1, com ataques tipo SYN *flood*, UDP *flood* e ICMP *flood*, e na Base de Dados 2, com o ataque DDoS Mirai. A Tabela 21 apresenta um resumo comparativo entre os resultados da Base de Dados 1 (KALI) e da Base de Dados 2 (MIRAI).

Tabela 21- Comparativo dos Resultados da Base de Dados 1 e 2.

Resultados por Base de Dados	Base de Dados 1: KALI	Base de Dados 2: MIRAI
Melhores Resultados em Medida F1	<ol style="list-style-type: none"> 1. EM com o Tempo entre chegada dos Pacotes: 96.50% 2. EM com as Três Características: 95.40% 3. Todos com o Tamanho do Pacote, com a exceção do K-Means (man) : 95.30% 	<ol style="list-style-type: none"> 1. K-Means (man) com o Tempo entre chegada dos Pacotes: 99.48% 2. K-Means (euc) com o Tempo entre chegada dos Pacotes: 99.46% 3. DBSCAN com o Tempo entre chegada dos Pacotes: 99.22%
Melhores Resultados em VP e FN	<ol style="list-style-type: none"> 1. Todos com o Tamanho do Pacote, com a exceção K-Means(man). VP 99.96% e FN 0.04% 2. EM usando as Três Características VP 99.95% e FN 0.05% 3. Farthest-First com o Tempo entre chegada dos Pacotes VP 99.91% e FN 0.09% 	<ol style="list-style-type: none"> 1. Farthest-First com o Tamanho dos Pacotes: VP 99.99% e FN 0.01% 2. Farthest-First com o Tempo entre chegada dos Pacotes: VP 99.93% e FN 0.07% 3. CANOPY com o Tempo entre chegada dos Pacotes: VP 99.88% e FN 0.12%
Melhores Resultados em VN e FP	<ol style="list-style-type: none"> 1. EM com o Tempo entre chegada dos Pacotes: VN 96.50% e FP 3.50% 2. K-Means(man) com o Tempo entre chegada dos Pacotes: VN 94.10% e FP 5.90% 3. EM usando as Três Características: VN 72.10% e FP 27.90% 	<ol style="list-style-type: none"> 1. EM com o Tempo entre chegada dos Pacotes: VN 99.98% e FP 0.02% 2. DBSCAN com o Tempo entre chegada dos Pacotes: VN 99.76% e FP 0.24% 3. EM usando as Três Características: VN 99.63% e FP 0.37%

Fonte: Elaborado pelo Autor.

Na Tabela 21 é possível observar os melhores resultados em termos de Medida F1, VP e FN ,e, VN e FP, para as Base de Dados 1 e 2. A detecção de ataques de negação de serviço pode ter requisitos variáveis em termos das métricas de avaliação de desempenho apresentadas. Em termos gerais, tende-se a buscar uma taxa elevada de detecção correta dos pacotes, balanceada com uma taxa reduzida de falsos alarmes e erros de detecção.

Para isso, utiliza-se os resultados da medida F1 como um equilíbrio entre as métricas de desempenho. Desse modo, é possível observar na Tabela 21 os atributos e métodos para o processo não supervisionado com maiores valores em termos de Medida F1 para cada um dos cenários apresentados.

Entretanto, sistemas de detecção de ataques de negação de serviço podem ser sensíveis ao alto número de falsos alarmes. Desse modo, são necessários combinações de atributos e métodos que tenham altos valores de Verdadeiro Negativo e baixos valores de Falso Positivo, o que possibilita um sistema com poucos falsos alarmes e onde poucos pacotes normais serão classificados incorretamente como pacotes de ataques. Assim, a Tabela 21 indica, para cada um dos cenários de ataque, as combinações de atributos e métodos com maiores valores de VN e menores valores de FP.

De outro modo, aplicações específicas de sistemas de detecção de ataques podem necessitar priorizar métodos que tenham altos valores de Verdadeiro Positivo e baixos valores de Falso Negativo, o que possibilitaria uma alta taxa de ataques detectados e que poucos ataques não serão detectados. Assim, a Tabela 21 também aponta quais seriam as características de rede e os métodos a serem utilizados para este caso.

Portanto, demonstra-se que para ambos os cenários de experimentos apresentados, os métodos de aprendizagem não supervisionados possibilitam realizar a detecção dos ataques de negação de serviços, e podem ser utilizados de acordo com os requisitos mais importantes em diferentes aplicações. Assim, tem-se que cada método um dos métodos de AM analisados são capazes de realizar a detecção dos ataques e podem ser recomendados para casos diferentes de detecção.

5. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Nesse trabalho de conclusão de curso foram apresentados o desempenho dos algoritmos de aprendizado de máquina não supervisionados na detecção dos ataques de negação de serviço no contexto da Internet das Coisas. Foram realizados experimentos em dois cenários reais contendo ataques de negação de serviço, os quais geraram duas bases de dados, nas quais os algoritmos tiveram seu desempenho testado e avaliado diante de métricas apresentadas nesse trabalho.

Para o Cenário 1, com ataques do tipo *SYN flood*, *UDP flood* e *ICMP flood*, foi possível obter 96.50% em termos de Medida F1 e para o Cenário 2, com o ataque Mirai, foi possível obter 99.48% na Medida F1. Além disso, conforme apresentado na secção 4.3, em ambos os cenários apresentados, foi possível obter valores elevados em termos de detecção de pacotes de ataque (VP) e pacotes normais (VN): Cenário 1 (VP 99.96% e VN 96.50%) e o Cenário 2 (VP 99.99% e VN 99.98%). Também, obteve-se taxas significativamente reduzidas de falsos alarmes (FP) e de classificação incorreta de tráfego de ataque (FN): Cenário 1 (FP 3.50% e FN 0.04%) e Cenário 2 (FP 0.02% FN 0.01%).

Além disso, esse trabalho de conclusão de curso apresentou a performance das características do tráfego de rede como atributos para o aprendizado de máquina. Dessa forma, a metodologia de aprendizado não supervisionada apresentou ser capaz de detectar ataques de negação de serviço em redes IoT, com apenas uma característica de rede, como o Tempo entre chegada dos Pacotes e o Tamanho do Pacote. Portanto, a sua utilização em sistemas de detecção de intrusão, poderá ajudar a reduzir os danos causados pelos ataques DoS em redes IoT.

Como trabalho futuro a este, tem-se realizado um estudo mais aprofundado sobre a seleção das características do tráfego de rede como atributos de EM para a realizar a detecção e classificação de ataques de negação de serviço, o qual será publicado em breve.

Além disso, sugere-se também como trabalho futuro, a implementação dos algoritmos não supervisionados apresentados no contexto de detecção de ataques de negação de serviço em arquiteturas que possibilitem a realização da mitigação da ação dos agentes maliciosos dado a detecção por esses métodos.

REFERÊNCIAS

ARAÚJO,A. "Uma Arquitetura utilizando Algoritmo Genético Interativo e Aprendizado de Máquina aplicado ao Problema do Próximo Release". 2015. Disponível em: <<https://www.researchgate.net>>. Acesso em 03 dez.2018.

ASHTON, Kevin. **That 'Internet of Things' Thing**: In the real world, things matter more than ideas. 2009. Disponível em: <<https://www.rfidjournal.com/articles/view?4986>>. Acesso em: 06 nov. 2018.

BENGHOZI, Pierre-Jean; BUREAU, Sylvain; MASSIT-FOLLÉA, Françoise. Defining the Internet of Things. In: BENGHOZI, Pierre-Jean; BUREAU, Sylvain; MASSIT-FOLLÉA, Françoise. **L'INTERNET DES OBJETS**. Paris: Éditions de La Maison Des Sciences de L'homme, 2009. cap. 1, p. 91-99.

BUCZAK ,A. L, E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection", *IEEE Communications Surveys & Tutorials* 18.2, 2016.

CHEN, Y.; Ma,X.; Wu,X. "DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory", *IEEE Commun. Lett.*, vol. 17, no. 5, pp. 1052-1054, 2013.

CLOUDFLARE,Cloudflare.com. **What Is The Internet Of Things (IoT)?**. Disponível em: <<https://www.cloudflare.com/learning/ddos/glossary/internet-of-things-iot/>>. Acesso em: 06 nov. 2018.

COMMUNITY IOT ONE. **5 Things you should know about MQTT**. Disponível em: <<https://community.iotone.com/t/5-things-you-should-know-about-mqtt/55>>. Acesso em: 06 nov. 2018.

CORERO. Mirai Botnet DDoS Attack Type: What is Mirai Botnet?. Disponível em: <<https://www.corero.com/resources/ddos-attack-types/mirai-botnet-ddos-attack.html>>. Acesso em: 06 nov. 2018.

COSEETTI,M. C. 2018. Kaspersky Lab. "Brasil é líder em ataques a dispositivos IoT".Disponívelem:<<https://tecnoblog.net/256165/brasil-e-lider-em-ataques-a-dispositivos-iot-com-30-mil-infectados-em-2018/>>. Acesso em: 20 de Agosto de 2018

DHANABAL, Dhanabal, L. e S. P. Shantharajah. "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms." (2015).

DOSHI, R. N; Apthorpe, N.; Feamster,N. "Machine learning DDoS Detection for Consumer Internet of Things Device", IEEE Security and Privacy Workshops(SPW), 2018

DROMARD, J. G. Roudière and P. Owezarski, "Online and Scalable Unsupervised Network Anomaly Detection Method," in *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 34-47, March 2017.

EVANS, Dave. **Cisco IBSG**: How the Next Evolution of the Internet Is Changing Everything. 2011. Disponível em: <https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf>. Acesso em: 06 nov. 2018.

GADDAM, S. R., V. V. Phoha, and K. S. Balagani, "K-means+ ID3: A novel method for supervised anomaly detection by cascading k-means clustering and ID3 decision tree learning methods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 3, pp. 345–354, 2007.

GHAFIR, K. G. Kyriakopoulos, F. J. Aparicio-Navarro, S. Lambotharan, B. Assadhan and H. Binsalleeh, "A Basic Probability Assignment Methodology for Unsupervised Wireless Intrusion Detection," in *IEEE Access*, vol. 6, pp. 40008-40023, 2018.

HOCHBAUM, Dorit S., David B. Shmoys, A Best Possible Heuristic for the k-Center Problem, *Mathematics of Operations Research*, v.10 n.2, p.180-184, May 1985

JUN,J.; Oh,H;Kim,S;. "Ddos flooding attack detection through a step-by-step investigation". IEEE International Conference on Networked Embedded Systems for Enterprise Applications, 2011.

JYOTHI,V.; Wang,X.; Addepalli,S. K.;Karri, R. "Brain: Behavior based adaptive intrusion detection in networks: Using hardware performance counters to detect ddos attacks", 2016 29th International

KOLIAS, C.; Kambourakis G.; Stavrou,A; Voas,J. "DDoS in the IoT: Mirai and Other Botnets", IEEE Computer, vol. 50, no. 7, pp. 80-84, 2017.

KURAMA,Vihar. Towards Science Today "Unsupervised Learning With Python". Disponível em : <wardsdatascience.com/unsupervised-learning-with-python>. Acesso em 02 dez.2018.

LOPEZ, M.I. & Luna, José María & Romero, Cristóbal & Ventura, Sebastian. (2012). Classification via clustering for predicting final marks based on student participation in forums. Proc. of 5th Int. Conf. on Educational Datamining. 148-151.

MCCALLUM, A., K. Nigam, L.H. Ungar: Efficient Clustering of High Dimensional Data Sets with Application to Reference Matching. In: Proceedings of the sixth ACM SIGKDD international conference on knowledge discovery and data mining ACM-SIAM symposium on Discrete algorithms, 169-178, 2000.

MITCHELL,T. M. Machine learning, 7th ed. NY, McGraw Hill, 1997

ÖZGUR, A., Erdem H. (2016) A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. *PeerJ Preprints*

PERVEZ,M. S.;Farid,D. M.;"Feature Selection and Intrusion Classification in NSL-KDD CUP 99 Dataset Employing SVMs", The 8th International Conference on Software Knowledge Information

RFID INC. RFID TAGS AND HARDWARE. Disponível em:
<<https://www.rfidinc.com/>>. Acesso em: 06 nov. 2018.

SAMUEL, A.L . Some studies in machine learning using the game of checkers. **IBM Journal of R&D**, v. 3, n. 3, p. 210-229, jul. 1959.

SAYAD,Saed. **K-Means Clustering.** Disponível em:
https://www.saedsayad.com/clustering_kmeans.htm. Acesso em: 06 nov. 2018.

SHALEV-SHWARTZ ,S e Ben-David S (2014) Understanding machine learning: from theory to algorithms. Cambridge University Press, Cambridge.

SHANTHAMALLU,, U. S., Spanias, A., Tepedelenlioglu, C., & Stanley, M. (2017). A brief survey of machine learning methods and their sensor and IoT applications. 2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)

TAN, P.-N.; STEINBACH, M.; KUMAR, V. Introdução ao datamining: mineração de dados. [S.I.]: Ciência Moderna, 2009.

TAVALLAEE, M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the kdd cup 99 data set. In: Proceedings of the Second IEEE international conference on Computational intelligence for security and defense applications, IEEE Press, Piscataway, NJ, USA, CISDA'09, pp 53–58.

VAZ, Arthur Lamblet. **Model Validation—Data Science.** Disponível em:
<https://medium.com/@arthurlambletvaz/model-validation-data-science-3084bb3a4ff8>. Acesso em: 06 nov. 2018.

YOUNG, Mark. “Improved Outcomes, manhattan distance”. 2018. Disponível em <http://www.improvedoutcomes.com> Acesso em 03. dez 2018.

ZHONG, S., T. M. Khoshgoftaar, and N. Seliya, “Clustering-based network intrusion detection,” International Journal of reliability, Quality and safety Engineering, vol. 14, no. 02, pp. 169–187, 2007.